Henderik A. Proper
Luise Pufahl
Dimka Karastoyanova
Marten van Sinderen
João Moreira (Eds.)

LNCS 14367

Enterprise Design, Operations, and Computing

27th International Conference, EDOC 2023
Groningen, The Netherlands, October 30 – November 3, 2023
Proceedings

THE ENTERPRISE COMPUTING CONFERENCE
EDOC2023 CONFERENCE

Springer

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.

Hend*erik* A. Proper · Luise Pufahl ·
Dimka Karastoyanova · Marten van Sinderen ·
João Moreira
Editors

Enterprise Design, Operations, and Computing

27th International Conference, EDOC 2023
Groningen, The Netherlands, October 30 – November 3, 2023
Proceedings

Springer

Editors
Hend*erik* A. Proper
TU Wien
Vienna, Austria

Luise Pufahl
Technical University of Munich
Heilbronn, Germany

Dimka Karastoyanova
University of Groningen
Groningen, The Netherlands

Marten van Sinderen
University of Twente
Enschede, The Netherlands

João Moreira
University of Twente
Enschede, The Netherlands

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-46586-4 ISBN 978-3-031-46587-1 (eBook)
https://doi.org/10.1007/978-3-031-46587-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

EDOC 2023 is the 27th conference in the EDOC series, which provides a key forum for researchers and practitioners in the field of Enterprise Design, Operations, and Computing. The EDOC conference series addresses the full range of models, methodologies, and engineering technologies contributing to building and evolving (intra- and inter-) enterprise systems across the "business-to-IT stack".

This year's conference places special emphasis on the theme of "Sustainable Enterprises", which reflects global developments toward a more sustainable society. In this context, new concepts and approaches are required: (a) to address pressing challenges ensuing from the need for sustainability in domains like healthcare, advanced manufacturing, finance, environmental management, agriculture, supply chain management, energy, and others, while ensuring long-term interoperability and resilience of enterprise systems; and (b) to leverage or rethink technologies such as digital twins, machine learning, IoT, big data analytics, distributed ledgers/blockchain, as well as novel computing approaches, when addressing the multifaceted nature of sustainable enterprise systems.

Following the new tradition started in 2022, the EDOC proceedings are published by Springer in the Lecture Notes in Computer Science (LNCS) series. This year's proceedings include 12 full papers selected out of the 36 full papers that were sent out for peer review (33% acceptance rate). All submissions were thoroughly reviewed in a single-blind process by at least three program committee members. Where reviewers had differing views regarding the qualities of papers, the program committee chairs initiated discussions among the involved reviewers to reach a consensus. The selected papers cover topical areas such as Enterprise Modeling, Enterprise Architecture and Engineering, Model-Based Software Engineering, Enterprise Analysis with Process Mining, Process Improvement and Engineering, and Modeling in an Enterprise Context. This year's EDOC program also includes two interesting Journal First presentations. We would like to show our greatest appreciation to the submitting authors and to the members of the program committee as well as additional reviewers for their hard work towards ensuring the high quality of the review process.

The proceedings furthermore include abstracts which pair with the invited talks of Coral Calero, Jerker Delsing, and Janina Bauer from Celonis; and with the tutorial offered by Marco Comuzzi and Paul Grefen. We would like to thank them all for their generosity in joining us in Groningen and their insights on the exciting and timely topics they present.

A separate companion post-conference proceedings will be published in the LNBIP series of Springer, with the papers from the EDOC Forum alongside workshop papers, doctoral symposium papers and demonstration track papers.

This year, EDOC is, for the second time, co-located with the International Conference on Cooperative Information Systems (CoopIS). Once again, we look forward to a most fruitful interaction between the involved communities, and an exciting overall program.

Finally, we would also like to thank the EDOC steering committee for entrusting us with the responsibility of organizing this year's conference. We would like to express our gratitude to all other members of the organizing committee, and in particular our local organization committee who have put in a lot of energy towards a successful event. As there can be no conference without engaged participation, we would like to thank all those who contributed with their insights to making our conference program interesting.

September 2023

Hend*erik* A. Proper
Luise Pufahl
Dimka Karastoyanova
Marten van Sinderen
João Luiz Rebelo Moreira

Organization

General Chairs

Dimka Karastoyanova	University of Groningen, The Netherlands
Marten van Sinderen	University of Twente, The Netherlands

Program Committee Chairs

Hend*erik* A. Proper	TU Wien, Austria
Luise Pufahl	Technical University of Munich, Germany

Steering Committee Chair

Selmin Nurcan	University Paris 1 Panthéon-Sorbonne, France

Steering Committee

João Paulo A. Almeida	Federal University of Espírito Santo, Brazil
Colin Atkinson	University of Mannheim, Germany
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Georg Grossmann	University of South Australia, Australia
Sylvain Hallé	Université du Québec à Chicoutimi, Canada
Zoran Milosevic	Deontik, Australia
Stefanie Rinderle-Ma	University of Vienna, Austria
Marten van Sinderen	University of Twente, The Netherlands
Giancarlo Guizzardi	University of Twente, The Netherlands

Workshop Chairs

Tiago Prince Sales	University of Twente, The Netherlands
Sybren de Kinderen	Eindhoven University of Technology, The Netherlands

Demo Track Chairs

Fadi Mohsen	University of Groningen, The Netherlands
Stefan Schoenig	University of Regensburg, Germany

Industrial Chairs

Zoran Milosevic	Deontik/Best Practice Software, Australia
Pascal Ravesteijn	HU University of Applied Sciences, Utrecht, The Netherlands

Proceedings Chair

João Luiz Rebelo Moreira	University of Twente, The Netherlands

Website Chairs

Magda Piekorz	University of Groningen, The Netherlands
Mirela Riveni	University of Groningen, The Netherlands

Publicity Chair

Mirela Riveni	University of Groningen, The Netherlands

Local Organization Committee

Ineke Schelhaas	University of Groningen, The Netherlands
Dimka Karastoyanova	University of Groningen, The Netherlands
Mirela Riveni	University of Groningen, The Netherlands
Fadi Mohsen	University of Groningen, The Netherlands

Program Committee

Agnes Koschmider	University of Bayreuth, Germany
Alan Wee-Chung Liew	Griffith University, Australia
Alexander Knapp	Augsburg University, Germany
Alfred Zimmermann	Reutlingen University, Germany
Amin Beheshti	Macquarie University, Australia

Andrea Marrella	Sapienza University of Rome, Italy
Andreas L. Opdahl	University of Bergen, Norway
Andrew Berry	ResMed Inc., Australia
Aniruddha Gokhale	Vanderbilt University, USA
Artem Polyvyanyy	The University of Melbourne, Australia
Asif Qumer Gill	University of Technology Sydney, Australia
Axel Korthaus	Swinburne University of Technology, Australia
Barbara Weber	University of St. Gallen, Switzerland
Bas van Gils	Strategy Alliance, The Netherlands
Ben Roelens	Open Universiteit, The Netherlands, and Ghent University, Belgium
Benjamin Yen	The University of Hong Kong, China
Carlos Azevedo	Federal Institute of Espírito Santo, Brazil
Chiara Di Francescomarino	DISI - University of Trento, Italy
Christian Huemer	TU Wien, Austria
Christian Zirpins	Karlsruhe University of Applied Sciences, Germany
Claudenir M. Fonseca	University of Twente, The Netherlands
Claudio Di Ciccio	Sapienza University of Rome, Italy
Colin Atkinson	University of Mannheim, Germany
Cristine Griffo	Free University of Bozen-Bolzano, Italy
Dominik Bork	TU Wien, Austria
Emilio Sulis	University of Turin, Italy
Estefanía Serral	KU Leuven, Belgium
Fadi Mohsen	University of Groningen, The Netherlands
Fatih Turkmen	University of Groningen, The Netherlands
Fethi Rabhi	The University of New South Wales, Australia
Flavia Santoro	Rio de Janeiro State University, Brazil
Florian Matthes	Technical University of Munich, Germany
Francesca Zerbato	University of St. Gallen, Switzerland
Frank Leymann	University of Stuttgart, Germany
Frederik Gailly	Ghent University, Belgium
Georg Grossmann	University of South Australia, Australia
Georg Weichhart	Primetals, Austria
Giancarlo Guizzardi	University of Twente, The Netherlands
Giuseppe Di Lucca	University of Sannio (RCOST), Italy
Guido Governatori	Independent Researcher
Hans Weigand	Tilburg University, The Netherlands
Hans-Georg Fill	University of Fribourg, Switzerland
Hend*erik* A. Proper	TU Wien, Austria
Hiroshi Miyazaki	Keio University, Japan
Irene Vanderfeesten	KU Leuven, Belgium

Irina Rychkova	University Paris 1 Pantheon-Sorbonne, France
Jürgen Jung	Frankfurt University of Applied Sciences, Germany
Jaap Gordijn	Vrije Universiteit Amsterdam, The Netherlands
Jan Øyvind Aagedal	Equatex, Norway
João Luiz Rebelo Moreira	University of Twente, The Netherlands
João Paulo Almeida	Federal University of Espírito Santo, Brazil
John Mylopoulos	University of Ottawa, Canada
Jolita Ralyté	University of Geneva, Italy
José Raúl Romero	University of Cordoba, Spain
Julio César Nardi	Federal Institute of Espírito Santo, Brazil
Julius Köpke	Alpen-Adria-Universität Klagenfurt Institute for Informatics Systems, Austria
Lam Son Lê	VGU, Vietnam
Leo Poss	University of Regensburg, Germany
Ljiljana Brankovic	University of New England, USA
Lorenzo Rossi	University of Camerino, Italy
Luis Ferreira Pires	University of Twente, The Netherlands
Luise Pufahl	Technical University of Munich, Germany
Madhushi Bandara	University of Technology Sydney, Australia
Manfred Reichert	University of Ulm, Germany
Manuel Wimmer	Johannes Kepler University Linz, Austria
Marco Montali	Free University of Bozen-Bolzano, Italy
Maria Teresa Gómez López	University of Seville, Spain
Maria-Eugenia Iacob	University of Twente, The Netherlands
Marten van Sinderen	University of Twente, The Netherlands
Martin Käppel	University of Bayreuth, Germany
Mathias Weske	HPI, University of Potsdam, Germany
Mattia Fumagalli	Free University of Bozen-Bolzano, Italy
Michael Rosemann	Queensland University of Technology, Australia
Michael Schrefl	University of Linz, Austria
Monique Snoeck	KU Leuven, Belgium
Nicolas Herbaut	Université Paris 1 Panthéon-Sorbonne, France
Oscar Pastor	Universidad Politécnica de Valencia, Spain
Paolo Ceravolo	University of Milan, Italy
Paul Grefen	Eindhoven University of Technology, The Netherlands
Peter Bernus	Griffith University, Australia
Peter F. Linington	University of Kent, UK
Peter Fettke	German Research Center for Artificial Intelligence (DFKI) and Saarland University, Germany

Pierluigi Plebani	Politecnico di Milano, Italy
Rüdiger Pryss	University of Würzburg, Germany
Rainer Schmidt	Munich University of Applied Sciences, Germany
Rajeev Raje	IUPUI, USA
Remco Dijkman	Eindhoven University of Technology, The Netherlands
Renata Guizzardi	University of Twente, The Netherlands
Ronny Seiger	University of St. Gallen, Switzerland
Ruth Breu	Research Group Quality Engineering, Italy
Sagar Sunkle	Tata Consultancy Services, India
Said Assar	Institut Mines-Télécom Business School, France
Schahram Dustdar	TU Wien, Austria
Selmin Nurcan	Université Paris 1 Panthéon-Sorbonne, France
Sharmistha Dey	Griffith University, Australia
Simon Hacks	Stockholm University, Sweden
Stefan Schönig	University of Regensburg, Germany
Stefan Tai	TU Berlin, Germany
Stefanie Rinderle-Ma	Technical University of Munich, Germany
Sybren de Kinderen	Eindhoven University of Technology, The Netherlands
Sylvain Hallé	Université du Québec à Chicoutimi, Canada
Tiago Prince Sales	University of Twente, The Netherlands
Ulrich Frank	Universität Duisburg-Essen, Germany
Uwe Zdun	University of Vienna, Austria
Vinay Kulkarni	Tata Consultancy Services Research, India
Wolfgang Maass	Saarland University, Germany
Yigal Hoffner	Shenkar College of Engineering and Design, Israel
Zoran Milosevic	Deontik, Australia

Additional Reviewers

Dhaini, Mahdi
Huynh, Tri
Maier, Pierre
Morais, Gabriel
Schumann, Felix
Guy, Ed
Lehner, Daniel
Middelhuis, Jeroen
Pettinari, Sara

Keynote Speeches

Always Look on the Green Side of Software

Coral Calero Muñoz

University of Castilla-La Mancha, Spain
coral.calero@uclm.es

That software moves the world is a clear fact. And that it is becoming more and more important, too. There are three aspects that have led to an increase in the intensity with which software is used: the Internet and social networks, data and artificial intelligence.

However, not everything is positive in the support that software provides to our daily lives. There are estimates that ICT will be responsible for 20% of global energy consumption by 2030, part of which will be due to software. And precisely the three mentioned aspects require large amounts of energy.

In this keynote we will review different concepts related to software sustainability, and we will show some results of software consumption measurements that we have carried out on the one hand, cases carried out to raise awareness in society in general about the impact that software has on the environment through examples of internet software and social networks; On the other hand, examples related to software, data and artificial intelligence carried out with the aim of creating a set of best practices for the software professionals.

Our ultimate goal is to make you aware of the consumption problem associated with software and to ensure that, if at first, we were concerned with the "what" and later with the "how", now it is time to focus on the "with what".

Flexible Production Value Networks a Possibility of an Utopia

Jerker Delsing

Luleå University of Technology, Sweden
Jerker.Delsing@ltu.se

Efficient and flexible production value networks is important for any producing industry. The currently obvious approach is digitalisation. Major complicating factors of cost efficient digitalisation is information interoperability. Information/data interoperability is a complex and non-trivial field. The is true for all type of information/data from enterprise level to production shop floor level. Addressing this situation will require both new technologies but even more a capability to implement such new technology.

In my keynote I will discuss the technology situation and new approaches to provide autonomous information/data interoperability in flexible production value networks. Implementing such new technology in production will require changes in investments and organisation strategies.

Can Processes Save the World?

Janina Bauer

Celonis, Germany
j.bauer@celonis.com

Processes determine the course of our lives: in our everyday routine or in companies - everywhere it is important when something takes place and how. What positive and negative influence do processes have on us? And what role do they play for companies and society in crucial transformations such as the sustainability transformation? The keynote looks at this from a solution-oriented, data-driven and practical perspective. We zoom in: From macroeconomic challenges and the carefully considered balance between sustainability goals and other economic goals, we analyze the supply chain and its processes as the biggest lever for sustainability success and go down to why every single activity's carbon footprint matters. We review the concept of Process Mining to create process transparency and automatize carbon emission measurements - but most importantly find emission hotspots and eliminate them to operationalize sustainability goals. And we are looking at the frontier of innovation and what has to come next. But for a really credible sustainability transformation we have to ensure that process transparency itself is a blessing rather than a curse. What does responsible Process Mining look like?

Blockchain for Business: Understanding Blockchain and How It Creates Business Value (Tutorial)

Marco Comuzzi[1] and Paul Grefen[2]

[1] Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea
mcomuzzi@unist.ac.kr
[2] Eindhoven University of Technology, The Netherlands
P.W.P.J.Grefen@tue.nl

While most people may know about blockchain from cryptocurrencies, blockchain is a technology that increasingly permeates the way in which modern businesses operate. However, its dynamics and functioning remain obscure to many. The objective of this tutorial is to provide the tools to understand the full extent to which blockchain technology is or can be used in business.

The first part of the tutorial focuses on the functioning of blockchain systems. Basic concepts, such as transactions, consensus mechanisms and smart contracts, are introduced initially at a conceptual level. The objective is to provide an implementation-agnostic model of blockchain that helps understanding existing solutions and future evolutions of blockchain. To maintain a solid grounding on real-world applications, we also discuss the instantiation of typical blockchain mechanisms into existing blockchain platforms, such as Bitcoin, Ethereum, or Hyperledger Fabric.

The second part of the tutorial focuses on the business applications of blockchain. We discuss decision-making tools to understand and assess the suitability of blockchain in different business scenarios and techniques to create business models that exploit blockchain. Examples and case studies of blockchain business applications are discussed extensively. Particular attention is given to the outcome management business scenario. This is a novel way of doing business in which providers are remunerated, instead of for the sale of products or services, based on the ability of the provided solution to create value for the customers. We show how blockchain is a key enabler of such scenarios, providing the underlying trust-building mechanism for verifying the value co-created by providers and customers.

Both parts of this tutorial include an interactive part, in which the opinion and expertise of the audience is sought to assess the potential of blockchain in different business scenarios.

The content of this tutorial is drawn from a book recently published by the speakers.

Contents

Enterprise Modeling, Architecture and Engineering

A System Core Ontology for Capability Emergence Modeling

Rodrigo F. Calhau[1,2,3](✉), Tiago Prince Sales[3], Ítalo Oliveira[3], Satyanarayana Kokkula[4], Luís Ferreira Pires[3], David Cameron[5], Giancarlo Guizzardi[3], and João Paulo A. Almeida[1]

[1] Ontology & Conceptual Modeling Research Group, Federal University of Espírito Santo, Vitória, Brazil
jpalmeida@ieee.org
[2] LEDS, Federal Institute of Espírito Santo, Serra, Brazil
[3] Semantics, Cybersecurity & Services, University of Twente, Enschede, The Netherlands
calhau@ifes.edu.br, {t.princesales,i.j.dasilvaoliveira,l.ferreirapires,g.guizzardi}@utwente.nl
[4] Department of Science and Industry Systems, University of South-Eastern Norway, Kongsberg, Norway
satyanarayana.kokkula@usn.no
[5] SIRIUS Centre, Department of Informatics, University of Oslo, Oslo, Norway
davidbc@ifi.uio.no

Abstract. To properly understand organizational adaptation and innovation, it is critical to understand the emergence phenomenon, i.e., how the capabilities of a system emerge after changes. However, for this, we should be able to explain systems, their structure, behavior, and capabilities. In pursuit of an understanding of the emergence phenomenon and the nature of those new kinds of systems in organizations, we propose a well-founded system core ontology based on the Unified Foundational Ontology. The ontology is also grounded in system science definitions and disposition theories. For a more integrated explanation of emergence, the proposed ontology considers distinct perspectives of a system, such as its composition, structure, properties, and functions. In the end, we discuss the applications and implications of the proposed ontology on the enterprise architecture area and emergence modeling.

Keywords: system · emergence · ontologies · enterprise architecture

1 Introduction

In recent years, we experienced rapid development of information technologies, such as artificial intelligence and cloud computing, which stimulated the emergence of new kinds of social and technical systems within enterprises [6,22,27]. In this context, enterprises are compelled to constantly adapt and innovate to remain competitive. This requires certain organization-wide capabilities that do not exist in specific individuals but are the result of a complex phenomenon

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 3–20, 2024.
https://doi.org/10.1007/978-3-031-46587-1_1

through which capabilities *emerge* from the interaction of organizational parts. An organization with innovative professionals does not necessarily imply an innovative organization, since this depends on the kind of relationships and interactions among them.

Emergence is a complex phenomenon that cannot be explained by just one cause. It is a result of the way the system's parts are related, the properties of parts, relational properties, and constraints, among other factors [9,26,28,31]. This phenomenon has been studied by system science researchers since the rise of the General System Theory (GST) [1,9,12]. These researchers consider the notion of *system* present in distinct fields in order to identify common principles among various system types (e.g. physical, chemical, biological, and social systems) [11]. According to these authors, an enterprise can be seen as a socio-technical system composed of interrelated technical and social parts whose capabilities emerge from the interaction between its parts [17].

At the same time, the structure and capabilities of enterprises have been studied in the Enterprise Architecture (EA) discipline [8]. In a similar way, EA notations support the representation of enterprises, their parts, relationships, and behavior, which, in theory, could be used to model capabilities resulting from the emergence phenomenon. However, there are no guidelines on how to properly model emergence. For example, how should one structure a team into roles in a way that maximizes the overall performance of the team? Or what is the best combination of functions for a team to be more productive? Since it helps to explain the emergence phenomenon, system science can ground EA with its theoretical foundation to model systems and help answer these questions. As addressed by [34], ontologies can play a fundamental role in this task since system models concern a distinct paradigm in organizational context.

Ontologies can improve the *expressiveness* and *domain appropriateness* of EA notations, as is shown in [3,14]. According to [20], ontologies have been useful in the computing field for representing and formalizing the semantics of various types of artifacts. Many ontologies have been proposed to model different types of systems, such as systems-of-systems [5], enterprise systems [27], smart systems [2], cyber-physical systems [35], to name a few. All these system-related ontologies focus on solving technological and practical issues related to a specific context and generally lack a wide system concept understanding, failing in particular to address the emergence phenomenon.

In this paper, we contribute to bridging this gap by applying concepts from system science authors such as Bunge [10,12], authors from GST [1,9] and systems engineering (SE) [16] areas, by leveraging the contributions related to emergence from system science. In [13][1] we proposed an ontology-based language pattern to ArchiMate to represent capability emergence based just on human capabilities. This previous work was grounded in disposition theories [7,19], to explain the emergence of capabilities, without considering system distinctions, their parts, functions, and connections. In this paper, we consider not just human capability but the capabilities of distinct system types in organizations. We also consider the phenomenon of emergence from the perspective of systems, their

[1] Under review.

components, and their relations. In order to provide a comprehensive account of the emergence phenomenon in a system, we consider distinct perspectives, including its composition, structure, properties, and function, based mainly on Bunge's "systemist" model [11,12]. To properly account for these system distinctions in EA notations, we propose a system core ontology based on the Unified Foundational Ontology (UFO). The ontology aims to: (i) improve the capability emergence modeling (in EA notations) of socio-technical systems into enterprises and; (ii) facilitate the identification of capability emergence patterns by using some pattern recognition technique. To regard this, the proposed ontology is also grounded on theories of parts and parthood [18,29], other system ontologies, system models [10,11,16,25], literature reviews on the system concept [16], emergence explanations [21,26,28,31], and disposition theories [7,19].

This paper is structured as follows: Sect. 2 presents an overview of the literature related to system theory concepts; Sect. 3 presents the Unified Foundational Ontology, used to create the proposed system ontology presented in Sect. 4, which is the main result of this work; Sect. 5 discuss the application of the ontology in the EA context and its implications; Sect. 6 shows an application of the ontology in the Spotify case, and; Sect. 7 presents the related works, and Sect. 8 concludes with our final remarks.

2 Emergence and Systems

The concept of "system" is strongly associated with emergence. Very often, a system is defined as a whole composed of related parts that allow the emergence phenomenon. For this reason, comprehending systems is fundamental to an account of emergence. System's definitions have distinct perspectives and most often vary from one author to another [16]. Despite this variation, authors who worked on the system concept recently or those whose work traces back to the origin of the General-System Theory, such as [1,9,11], converge on an understanding of a system as a kind of 'complex' (or 'organized' whole) composed of 'connected' (or interacting) elements. In this sense, a system is understood as a collection of things that, through their connections (or interactions), creates something new, such as behavior or emergent properties [9,16]. Reinforcing this understanding, Dori and Sillitto [16] presented the following generic definition of a system based on the extensive literature review by analyzing more than one hundred definitions:

> *A system is a set (or combination, group, collection, arrangement, organization, assemblage, assembly, ensemble) of parts (or components, elements, objects, subsystems, entities) combined (or integrated, organized, configured, arranged) in a way that creates (or enables, motivates) properties (or functions, processes, capabilities, behaviors, dimensions) not possessed (or exhibited, presented) by the separated (or individual, single) parts (or components, elements, objects, subsystems, entities).*

Dori and Sillitto's generic definition captures the many terms used to refer to the different aspects of the complex notion of a system. It alludes to the inevitable *plurality* of things that in *combination* make up a system ("a set of parts combined...."), also referring to the existence of properties that characterize the system *as a whole*, i.e., beyond the properties of parts in isolation ("... in a way that creates properties not possessed by the separated parts").

Bunge [10,11] classifies system properties into *resultant* and *emergent*. Resultant properties are those that can be decomposed, explained, or reduced into properties of a system's parts. For example, the total mass of a system is defined by the simple sum of the masses of its components. Unlike resultant properties, emergent properties are those that, while related to the properties of parts, are not present in isolation in the separated parts. For example, the buoyancy of a ship cannot be reduced to the buoyancy of its parts (an arbitrary piece of a steel hull is typically not buoyant by itself). In the words of [28], the *emergent properties supervene* on the components' properties. According to [21,26], the emergent properties are also the result of *system constraints*, which limit it on the one hand but enable the arising of new characteristics on the other. [21] exemplifies this through the restrictions caused by the knee in the femur and tibia movements, which contribute to the emergence of the walking capability. In the same way, according to [31], *emergent properties* are a direct consequence of the relationships among parts. For instance, as the author exemplifies, what distinguishes diamond and graphite is the way that carbon molecules are associated. Based on these differences, distinct emergent properties appear as transparency, in the case of a diamond, and electric conductivity, in the case of graphite.

In system definitions, emergence is also associated with system functions. As [30] states, the concept of *function* is generally related to a teleological perspective on a system. In this sense, *functions* are manifested through some goal-oriented result or behavior (process, action) [1,15]. *Functions* are frequently defined as a kind of *property* related to the capability (or disposition) concept [15,27]. As they have a relational aspect, some authors, such as [15], also see *functions* as a kind of *role* played through behaviors that are required by the system's capability definition. In general, *functions* are seen as a system-dependent aspect, called "system function" by [15], inherent specifically to one bearer. However, according to the authors, *functions* can also be seen in a more "generic" way, independently of a specific system (called "ontological function" by them). In this case, these "generic" *functions* are useful for designers to describe an intended system, before building or acquiring it. As [15,30] state, initially, the system's *functions* description starts identifying the system *macro-function*, as it relates to the whole system. Then, as the authors describe, this *macro-function* is decomposed into *sub-functions* (often to be assigned to the system parts), a process known as (logical) *functional decomposition*.

As explained above, the emergence phenomenon results from related parts and not isolated ones. Bunge [11] defines system structure as a set of all relationships between a system's parts. According to [12], the relationships between system elements can be *"bonding"* or *"non-bonding"*. *Bonding* relationships (also termed *"connections"*) are those in which one element somehow causes changes and impacts the other [11]. These relationships can also be characterized by

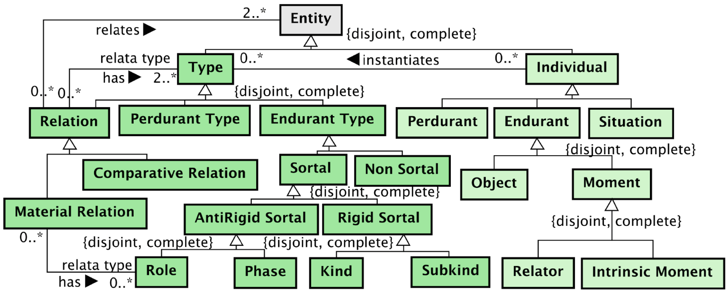


Fig. 1. Unified Foundational Ontology fragment

the flow of energy, material, or information between elements [11]. Otherwise, *non-bonding* relationships are those that do not impact the *relata*, such as comparative relationships as temporal or spatial relationships [12] (e.g., "higher than" or "younger than"). The set of the *bonding* relationships connecting all system's elements at a specific time shapes the system's *bonding structure* (or "configuration") [11]. In this context, the *bonding structure* represents the system's form, organization, or arrangement [16]. It allows the system elements to interact with others and, consequently, the whole system to display its behavior. [12] defends that an object needs a *bonding structure* to be considered a system. [1] remarks that each system's element must be connected to every other one, either directly or indirectly. Hence, in a system, there must be no isolated subset of elements. In this sense, not connected or independent elements should not be considered as a system [30].

3 Foundational Baseline

Foundation Ontologies are both Formal and Reference Ontologies. Guizzardi [20] proposes in his work the *Unified Foundational Ontology* which describes domain-independent and general concepts and that we will adopt in this work to perform ontological analysis. Figure 1 presents a UFO fragment that shows its fundamental distinctions, among *individuals* and *types*. These two concepts basically represent types and their instances. For example, *person* is a *type*, and Karl and John are *individuals*, instances of the *person* type.

UFO divides *individuals* into *endurants*, *situations*, and *perdurants* (or events). *Endurants* are *individuals* that persist in time, maintaining their identity (i.e. John, The Beatles, Spotify Technology S.A.). *Perdurants* are *individuals* that manifest themselves through time (e.g. John's birthday party, the inauguration of the Pope Francis). *Situations* are individuals composed (possibly) of many other individuals (including other situations) that may trigger *events*.

Endurants are divided into *objects* and *moments*. *Objects* are *endurants* that do not depend on another *individual* to exist (e.g., the Earth, John, an apple). In contrast, *moments* depend on another *individual* to exist (e.g., Mary's age, Gerald's headache, the reddish color of an apple). A *relator* is a specific type of *moment* capable of connecting two or more entities (e.g., a marriage that connects spouses). In contrast, an *intrinsic moment* is one that is existentially dependent on a single individual. Intrinsic moments include *qualities*, categorical properties such as color, height, weight, etc., and *modes*, which are moments that can bear their own moments and can vary independently. *Modes* include *dispositions*, which are moments that can be manifested through *events* in certain *situations*. Examples of dispositions include John's ability to speak English, and an airplane's flying capability. Based on UFO-C (extension of UFO approaching social aspects), *agents* are considered *objects* that perceive *events* and perform *actions* based on a background of *beliefs*, *desires*, and *intentions* (special categories of *intrinsic moments* termed *intentional moments*, omitted from the figure for brevity). As depicted in the model, *agents* can be *physical* (e.g., humans and animals) or *social* (e.g., teams, organizations, communities, etc.), and all of these are considered potential bearers of *capabilities* and *intentional moments*.

Although UFO does not include the concept of *system*, it includes the concept of *functional complex*, which is similar to the concept of *system* from a mereological perspective. Functional complexes are *objects* whose parts play distinct *functional roles* with respect to the *whole*. In this case, the *parthood relation* is defined in UFO by the *"is a component of"* relationship [20]. The *"is a component of"* relationship is a type of mereological relationship between *functional complexes* that establishes the part in a functional complex's whole. In this case, it establishes the *functional role* played by the *parts* (components) in the *functional complex*. For example, in a chair, each wooden piece has a different *functional role*: front leg, back legs, seat, etc. Besides not being the focus on [20], these *parts*, which play distinct *functional roles*, should be related between them. Besides that, UFO also considers another whole, namely *collectives*. Differently from *functional complexes*, collectives are wholes whose parts perform the same *role* type. Examples of collectives include a group of students performing a group assignment or a collection of books in a library.

Types are predicative entities whose instances share common features. In the taxonomy of types in UFO, there are *endurant types*, *perdurant types*, *object types*, etc., according to the ontological nature of their instances. *Types* also include *relations* between two or more *entities*. As shown in Fig. 1, relations in UFO are associated with two or more relata types. *Material relations* are those that apply in the presence of a *relator* mediating the *relata*, e.g., the "married with" relation requires a *marriage*, the "enrolled in" relation requires an *enrollment*, and so on. As illustrated in Fig. 1, *material relations* have *roles* as *relata type*. For example, the "marriage with" *material relation* has the spouses' roles as its *relata types*. *Comparative relations* are another kind of relation in UFO, as Fig. 1 depicts. They are called *formal relations* since they involve two or more entities directly, without the intervention of a mediator (e.g. "taller than" and "younger than" relations).

Guizzardi [20] also categorizes the types according to the identity principle that entities maintain. Based on this, *types* are classified as *rigid* and *anti-rigid*. *Rigid* types are those that apply necessarily to their instances through their whole existence, and include *kinds* and *subkinds* (e.g. "Person", "Car", "Pineapple"). In contrast, *Anti-rigid* types are those that classify their instances contingently (or "dynamically"). *Roles* are anti-rigid types whose contingent classification conditions are relational (e.g., "Student" and "Employee") [20]. UFO also considers *non-sortal* types representing common properties of individuals of multiple *kinds*: (i) *categories* subsuming multiple kinds rigidly (e.g. mammal); and (ii) *role mixins* which subsume various *roles* with distinct *kinds* (e.g., "Customer", subsuming "Personal Customer" and "Organizational Customer").

4 The System Core Ontology

For the ontology modeling, we use the OntoUML notation, proposed by Guizzardi [20], as a UML extension that addresses the foundational distinctions from UFO in UML class diagram through stereotypes. The ontology requirements were identified based on GST literature and in Bunge's Composition, Environment, Structure, and Mechanism (CESM) model [12]. Therefore, the system ontology must account for a: **R1) system's composition** concerning its *components*, *subsystems*, and their hierarchical relations; **R2) system's structure** concerning the notions of (internal and external) *connection* and *non-bonding relationship*; **R3) system's function** concerning the *roles* played by the system and components in a *functional decomposition*; and **R4) system's characterization** concerning the system's *emergence* phenomenon.

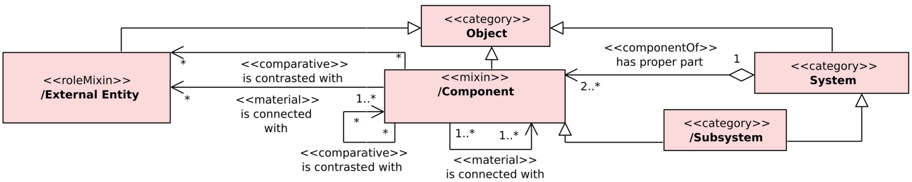


Fig. 2. Well-founded System Ontology (Composition and Structure)

Concerning system connectivity, *systems* are considered in this work as *complexes* of related elements [9] or *integral wholes* [29] since they need a "unifying condition" to exist. When it comes to UFO distinctions, we consider *systems* as being a *category* of *object*, as Fig. 2 depicts. By being an *object* in UFO distinctions, systems can also be (social or physical) *agents* with intentions, desires, beliefs, perceiving events, and performing actions. Based on this distinction, an organization or a team are considered a *social agent* and also (socio-technical) systems formed by other agents (e.g., teams, humans, etc.) and a non-agentive objects (e.g., equipment and other resources). Specifically, based on [20], systems are *functional complexes* since the system's components also perform different *functional roles* in the respective whole. In this case, as *functional complexes*, we consider the following criteria that differentiate *systems* from "simple" (atomic) *objects*: (i) the complexity degree (number of components and connections);

(ii) the integrated (bonding and non-bonding) structure formed by components' connections; (iii) the heterogeneity and complementarity of components' functional roles (and capabilities); and (iv) the emergence of new properties and new behaviors.

In relation to its *composition*, the *system* "has proper parts" called *components* (a derived class), as shown in Fig. 2. In this case, *components* represent *mixins* of interrelated *objects* (including *functional complexes*, *quantities*, and *collectives*), in UFO terms. In this ontology, they are generic concepts that correspond to all kinds of system's parts (or elements), such as *units*, *blocks*, *modules*, *interfaces*, *ports*, and even other *systems*, called *subsystems* (a derived class), as depicted in Fig. 2. These *components* could be arranged hierarchically, through the part-hood relation and they are interrelated. For example, the wooden stool *ws* has as proper parts the legs $l1$, $l2$, $l3$, and $l4$, besides the seat *st*.

We consider in the ontology the two senses of the "function" of the system (and components) addressed by [15]: one related to its "*position*" and the other to its "*capabilities*". Regarding the former, it is a more "generic" perspective, independent of a specific system, as [15] approach. In UFO terms, this perspective is closely related to *roles* (specifically *functional roles*), due to their generic, anti-rigid, and relational nature. As a result, in this sense, the *function* (in the sense of "position") of the system (or its component) is a *functional role* it instantiates, as depicted in Fig. 3. As shown, the *functional roles* are "characterized by" *moment types*. As depicted, *moment types* can "complement" (and also trigger or block) others, based on the same disposition theories above. In addition, based on [15], we also regard the (macro) *function* of a system can be decomposed into (sub) *functions*. This aspect is shown in Fig. 3 by the "constitutes" relationship between *functional roles*, forming a functional *role's* "hierarchical structure". In this case, to be a system, its sub-functional roles must be heterogeneous but also complementary, based on the *moment type's* complementary relations. Finally, the decomposition of a *functional role* and the parthood relationships between components and the system must match.

Concerning the second sense of *function*, as something intrinsic to a specific bearer, we consider that *systems* and their *components* bear *capabilities* (i.e., subtypes of *system moment* or *component moment* with dispositional nature) which can be specialized in this context to perform *functions*. So, in this sense, we mean *functions* as specialized *capabilities* (*dispositions*) that bring "benefits" to other entities in the system context, as defined by [16,24]. As illustrated in Fig. 3, for a *system* or *component* to perform some *functional role*, it must have certain *capabilities* (*moments*) in compliance with the *moment types* that characterize this *functional role*. For example, the wooden stool *ws* performs the stool (macro) *functional role* of "supporting a person"; the leg $l1$ performs the front-right leg *functional role*, $l2$ performs the front-left leg *functional role*, $l3$ performs the back-right leg *functional role*; $l4$ performs the back-left leg *functional role*, and the seat *st* performs the seating *functional role*. While performing these roles, the components must have specialized *capabilities* (functions): $l1$ must support seat *st*; $l2$ must support seat *st* too; seat *st* must support the person.

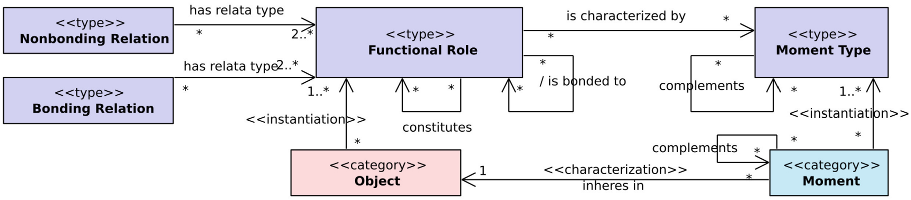


Fig. 3. Well-founded System Ontology (Functional Aspect)

In this work, we adopt the *bonding* and *non-bonding relations* definitions of Bunge [11,12], concerning that one relatum impacts the other in the *bonding relation* and in the *non-bonding relation* it does not. We regard these definitions from the two UFO perspectives: *types* and *individuals*. In the former, *bonding* and *non-bonding relations* correspond to "types of relations" between *functional roles* (types). This aspect is shown in Fig. 3, in which both *bonding* and *non-bonding relations* have two or more *functional roles* as their relata. Concerning these relations from the *individuals'* perspective, they follow the same principle. When the *functional roles* are instantiated by some entity (a system, its components, or external entities), the *bonding relation* type is "embodied" in (or instantiated by) these entities. This instantiation is illustrated in Fig. 2 by the "connected with" relationships between *components* and between *components* and *external entities*. In this context, *connections* between only components in a system are called *internal connections*, and *connections* between components and *external entities* are called *external connections*. In a *non-bonding relation*, as two or more *functional roles* are related as a kind of restriction, to instantiate this *functional role*, the *component* must satisfy these restrictions established by the *non-bonding relation*. The instantiated *non-bonding relation* is represented in Fig. 2 through the "is contrasted with" relationships between *components* and between *components* or *external entities*.

Concerning the ontological nature of *bonding relations*, when an *object* instantiates a *functional role*, its *dispositions* (capabilities) required for this role are specialized to attend to the system's peculiarities, in order to perform *functions*. So, for meeting the system's unifying condition, those "specialized" *dispositions* need to become externally (and existentially) dependent on other objects (components or external entities) and interact in the system context. As a consequence, relationships between connected objects are mediated by these externally dependent *dispositions* (capabilities). Based on this, we consider in this work that *bonding relation* is a kind of *material relation* in UFO terms. In contrast, Bunge's *non-bonding relations* are not *material relations* in UFO terms. Since they just regard constraints or restrictions, they are considered a kind of descriptive relationship that relies on the relata's intrinsic properties (UFO), unable to change the relata's state. As a result, they are related to the UFO *comparative relation* type. In the case of the wooden stool *ws*, the legs *l*1, *l*2, *l*3, and *l*4 are "connected with" the seat *st* (bonding relation); the legs "is contrasted with" each other since they are parallel, and they are perpendicular to the seat *st* (non-bonding relations).

Based on [10], systems have "global properties", not founded in their parts. We call them *system moments*, as shown in Fig. 4. In UFO terms, this work considers *system moments* as a *category* of *moments*. As a consequence, *system moments* represent *extrinsic moments* (e.g., "*relators*", "mutual properties") and *intrinsic moments*, as *qualities* (e.g., "attributes") and *modes* (e.g., dispositions and capabilities). We also adopted here the distinction of *emergent* and *resultant properties* addressed by [10]. As discussed, emergence is not a simple phenomenon that appears from one and only factor, but it is a result of the component's properties combination, constraints (represented by comparative relationships), and connections [21,26,28,31].

According to that, we consider in this work that an (emergent) *system moment* "emerges in" a specific *system situation*, as depicted in Fig. 4. In this case, the *system situation* represents the *components* of a *system* and their relationships in a certain occasion. Besides emerging "in a *situation*", *system moments* also "emerge from" (component) *moments*, as shown in Fig. 4 (including *dispositions*, *qualities*, *relators*). In a complementary way to the emergence explanations, we also consider the relationships between *moments* (i.e., *dispositions*) to explain emergence, as approached in [13]. As a result, for a *system moment* to emerge from (components) *moments*, the latter must be (inter) related. In this sense, based on disposition theories, we consider that components' moments can be complementary, i.e., reciprocal (mutually activated) [19] or additional (additionally activated) [7]. As shown in Fig. 4, both types of relations are considered in the "complements" relationship. We also consider which *component moments* can have relationships of triggering [7] and blocking [19] (not shown in the model as these relationships are derived). For example, the wooden stool *ws* has the "supporting capability" as an emergent *system moment* (emerged from the "supporting capability" from legs and seat). Concerning the disposition relation in this example, the "supporting capability" of each leg is additional to each other and they are mutually reciprocal to the "be supported capability" of the seat.

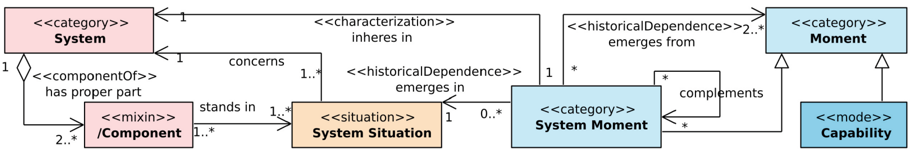


Fig. 4. Well-founded System Ontology (Capability)

5 Capability Emergence Modeling in Enterprise Architecture

As exemplified, a wooden stool must follow some "guidelines" in order to allow the emergence of its "comfortably supporting humans" capability and thereby fulfill its function. Based on the ontology distinctions, it must be composed of rigid pieces performing legs and seat roles; these roles must be complementary; these rigid pieces must attend quality criteria (e.g., have certain resistance); the legs must have the same height; must be parallel to each other and perpendicular to the seat. These guidelines are a "replicable" generalization that establishes patterns of emergence that a stool must follow.

Similarly to simple systems such as a stool, a socio-technical system such as a team, or an organization, also has certain "emergence patterns". For example, an organization can determine that productive teams are those in which developers (functional role) collaborate (connection); have complementary skills but share the same values (comparison); and the tech leader (functional role) supports the developers (connection) and is more experienced than them (comparison). In sum, a successful organization that evolves and adapts is one that is able to identify and replicate these "emergence patterns", remaining and creating new capabilities even when it changes.

The system's ontological distinctions can help with identification and representation of these patterns. They can facilitates the description of components their connections, properties, capabilities, functional roles, etc. These distinctions form guidelines that can be used to create: (i) *capability emergence models* for a particular system (e.g., a specific team), used to understand the emergence for create, change, or analyze it; or (ii) "general" *capability emergence patterns*, using *functional roles* to generalize the emergence phenomenon from different systems (e.g., all teams of an enterprise) and reply (specialize) it in distinct situations. Examples include appropriate combinations of (complementary) professional roles for building performative teams; the capability types each professional needs to have for product development with high quality; the types of relationships these professional roles need to have to allow collaboration; the more appropriate equipment type for a type of professional role to increase safety, and so on.

Embedded in these guidelines, the ontology also provides some "modeling patterns" to be satisfied by these *capability emergence models*, improving the system and emergence representation. In a *capability emergence model* for a particular team, for example, all team members must be connected (directly or indirectly); team members must satisfy constraints; team members must have complementary *capabilities* (reciprocal and additional); these related *capabilities* must follow the *connections* between the team's members; the *connections* must also happen just between team's members with related *capabilities*; team's members must have a function (*functional role*) in the team; and team members must satisfy the *functional role* criteria. Regarding the emergence patterns, it must satisfy the following principles for a team, for example, the team's macro-functional role must be constituted of at least two or more team member's sub-functional roles; all team member sub-functional roles must be related to *bonding relations* (directly or indirectly); team member sub-functional roles must be complementary/additional (based on the moment types); team member sub-functional roles must have constraints; all team member *functional roles* must be characterized by *moment types;* and non-bonding relations between the team's member functional represent restrictions to be satisfied.

These emergence patterns can be identified from the literature, success cases of organizations, experts, or even pattern recognition techniques which can analyze organizational data to identify these patterns. In this case, for example, the better team's structure (roles and relationships) and professional characteristics (desired capabilities for each role) can be identified from data of the more performative teams (with better KPIs) of an organization. To facilitate the identification and application of the emergence patterns, these tasks can be included into the knowledge management process of the organization. In this case, the identified patterns can even be considered as a knowledge items into the knowledge repository of the organization and forming a "library" of patterns.

Another important implication of this work is the improvement of EA notations through ontological analyses. As a result, it could be used as the basis for the creation of language patterns to increase the expressiveness of these notations, incorporating system-related concepts and emergence-related aspects into them. The use of the system ontology to create language patterns in EA notations will be detailed in the following case study.

6 Emergence in the Spotify Company

To show the benefits of the system ontology and the guidelines shown above, we have applied them to a real-life case study from the literature to improve the capability emergence modeling. This study case was addressed originally by Bäcklander [4] and concerns the Spotify company. Bäcklander [4] focuses specifically on understanding the emergence patterns followed by the company, such as (1) how adaptability and related capabilities (e.g., self-organization, learning, collaboration, etc.) emerge in the company, and (2) how the agile coach position contributes to these capabilities. Bäcklander [4] performed an ethnographic study inside the company, observing and interviewing the agile coaches. After the interviews, the author identified the main characteristics, practices, interactions, and motivations of the agile coaches that contribute to the emergence of these organizational capabilities.

The Spotify company is well-known for having a unique structure, which distinct authors describe [4]. Spotify is a socio-technical system composed basically of guilds and tribes. Guilds represent cross-cutting study groups focused on employee development and which anyone can join. In contrast, tribes are focused on the development of solutions. Tribes are composed of squads, a kind of development team in Spotify. Tribes are also formed by chapters in Spotify. They are "local" study groups that anyone can join, similar to guilds. In this case, while tribes and squads are result-oriented and highly coupled groups, guilds and chapters are learning-oriented and loosely coupled groups. Each squad has members (e.g., the developers) and is related to a product owner. Besides this, each tribe has its own agile coach supporting the developers of each squad.

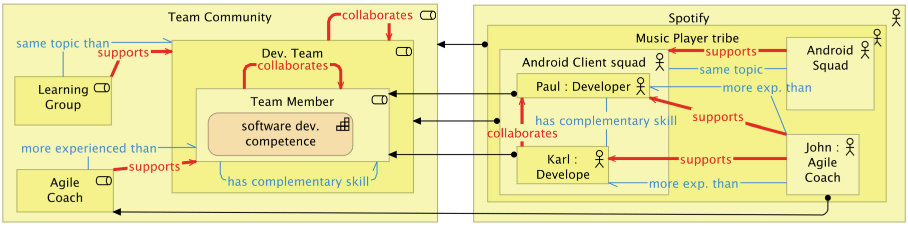


Fig. 5. Spotify as a System

To allow the emergence of the adaptability capability, Spotify considers some capability emergence patterns to be followed by the company, based on successful experiences. In this pattern, the company considers not only professional capabilities but also functions performed by them, and the connection between them, among other elements. These patterns are replicated in all subsystems (squads, guilds, etc.), at all levels of the organization, impacting the whole company. As a result, they contribute to creating a flexible organizational structure, a condition for the emergence of adaptability. Each of these components of Spotify mentioned above performs a certain *functional role* in the system: (i) *team community* functional role, played by "tribes"; (ii) *learning community* functional role, played by "guilds"; (iii) *development team* functional role, played by "squads"; and (iv) *learning group* functional role, played by "chapter". *Team community*, *learning community*, *development team*, and *learning group* are examples of *functional roles* played by social entities. They are constituted by *sub-functional roles* played by people, such as *agile coaches* and *developers* functional roles.

These functional roles also must be characterized by *capability types* and have *bonding* and *non-bonding relations* as part of Spotify's *emergence pattern*. For example, (i) learning groups (chapters) support development teams; (ii) development teams support other development teams; (iii) agile coaches support team members; (iv) team members collaborate with other team members, among others. Regarding non-bonding relations, they describe distinctions that should be attempted by Spotify's components, for example (i) learning groups (chapters) have the same topic as some team (squads); (ii) agile coaches must be "more experienced than" team members; (iii) team members "have complementary competence than" other team members; (iv) chapter member "has similar competence to" other chapter members, among others. Concerning the characterization of these *functional roles*: (i) the agile coach must have supporting, leader boosting, and communication skills; and (ii) the developer must have agile development competence and communication skills; Spotify and its components should instantiate these functional roles in order to replicate the emergence pattern and attempt the "unifying conditions" to form a whole (system).

To improve the representation of this emergence pattern, we created an illustrative language pattern in ArchiMate specific to this case, as shown in Fig. 5. As illustrated, *functional roles* are represented as BUSINESS ROLE construct (when it is performed by an agent) or RESOURCE construct (when it is performed by an object, as equipment for example) using the assignment relation; *capability types* are represented as CAPABILITY constructs related to *functional roles*; *bonding relations* and *connections* are represented using TRIGGERING, FLOW, or SERVING relationships between *functional roles* (highlighted in red); *non-bonding relations* and *comparisons* are represented using ASSOCIATION relations related to *functional roles* (in blue); and *system*, *component*, *external entities* are represented using STRUCTURAL ELEMENTS, related to functional roles through realization relation; and parthood relationships are represented using COMPOSITION or ASSOCIATION relationships.

The emergence phenomenon in Spotify is one of the main aspects explained in this case study, even because Bäcklander [4] considers complex and adaptive system theory as a foundation. As a result, Spotify is seen by the author as a complex system that belongs to a changeable environment. In this case, the work explains how agile coaches play a special position since they contribute to the emergence of some capabilities, especially the adaptability and evolution of Spotify. The author associates adaptation and evolution capabilities with learning, open dialogue, and creativity capabilities. According to the study case, these capabilities in Spotify are a result of the agile coaches acting as enabling leaders and creating adaptation spaces in the company.

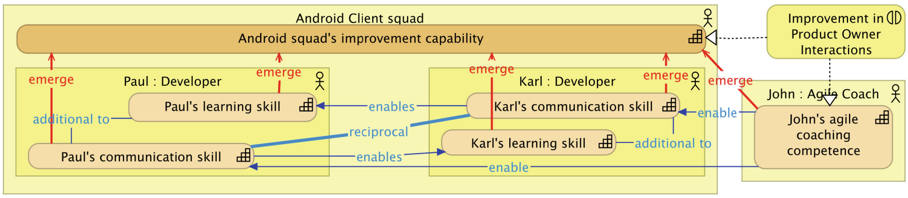


Fig. 6. Emergence phenomenon in Spotify

However, the author explains the influence of the agile coach in this case, but does not explain how emergence happens. As stated, the emergence in Spotify's context is a result of an emergence pattern that addresses relationships and properties to be satisfied by the professionals performing certain functional roles. In order to reach this pattern, those professionals establish connections between themselves, in accordance with the functional role relations, as detailed above. Based on these connections, the professionals are able to interact and perform tasks. These connections are a result of their complementary capabilities (additional and reciprocal). They are also a consequence of enabling relationships between capabilities.

Figure 6 illustrates how the agile coach contributes practically to the emergence phenomenon in a squad in this emergence pattern. As depicted, in this case, one of the main contributions of the agile coaches is enabling better communication between the squad members through the supporting connection. Consequently, the developers can share more of their opinions, among other communication skills. As it is shown, the more the (reciprocal) communication skills of the developers are enabled by agile coaches, the more their learning (and reflection skills) are enabled through these interactions, allowing them to create new solutions. In summary, as a result of these dynamics, the improvement capability of the squad emerges, as illustrated in Fig. 6. Therefore, as a result of the emergence of the improvement capability in each squad stimulated by the agile coach, the improvement capability of the tribe also emerges, contributing to the emergence of adaptability in the whole organization.

7 Related Work

One of the main related work in the context of our work is the Systemic Enterprise Architecture Methodology (SEAM) [34]. SEAM is concerned with a method for assisting in the modeling of businesses as complex systems. SEAM enterprise modeling addresses multiple levels that can guide emergence phenomenon modeling. Aside from that, the methodology is based on an ontology that addresses fundamental concepts such as object, action, state, location, time, space, and characteristics. Nonetheless, the SEAM method does not center on emergence phenomenon modeling nor on capabilities. Therefore, no modeling guidelines are provided in this case. Furthermore, the ontological distinctions fail to take into account basic concepts from system science such as system, function, component, connection, and other relationships. Otherwise, the present work considers these basic distinctions in a well-founded manner.

Concerning system ontologies, a number of them focus on *engineered systems* [25] as *cyber-physical systems* [6,35] and systems-of-systems [27]. Most of these ontologies focus on defining systems, components (subsystems), and their parthood relationships. A part of these models also considers system characteristics, such as attributes, properties, and capabilities [5,6,25]. However, almost none of the models consider the representation of the *emergent property*. Some exceptions [23,33] define an *emergent property* as a property that belongs to the whole system, not its components. Besides this, these models do not relate the *emergent properties* to the basic properties (or a kind of situation), which are inherent to the system parts.

System function is not a well-covered aspect in related work. Many of them [25,27,35] link the (whole) system to a "generic" function concept (more related to an intrinsic aspect of it). In addition, in these works, the functional decomposition of the system is not considered. Otherwise, some of the works [6] relate the system to a kind of role that it can perform (or position that it occupies). They also allow a hierarchical representation of the system's *functional roles*. As a result, these ontologies enable the functional decomposition representation, including not only the system's *functional role* as a unit but also the component's *functional role*. Most of the works [5,6,23,25,27,33] consider some kind of structural relationships among systems (or their components). Some of these works define this structural relation explicitly through a concept like *connection* [25], structural relation [27], connector, link [5], interaction, or binding mutual [33]. Others also consider some kind of mediators as connection points, ports, or interfaces [6]. One aspect not considered by the ontologies is the *non-bonding relationships* between systems and components. The only exception is in [33], which defines a kind of mutual property called non-binding property. Most of these system-related ontologies are focused on addressing technological and practical challenges in particular situations and, overall, lack a broad and comprehensive well-founded system notion, failing to deal with the emergence phenomenon.

8 Final Remarks

The complexity of the systems in society is increasing considerably as a consequence of technological development. It has given rise to new kinds of more complex and diverse systems. In this context, ontologies are crucial to a better understanding of these systems. To address this issue, this work aimed at proposing a well-founded *system ontology* based on Unified Foundational Ontology [20]. This ontology was proposed based on GST principles, allowing the broad representation of the distinct kinds of socio-technical systems including their composition, function, structure, and properties. The major implication for Enterprise Architecture of this ontology is to provide guidelines for capabilities emergence modeling and emergence pattern identification in EA notations (e.g., ArchiMate). This system ontology can also be used as a reference to integrate ontologies of distinct kinds of systems such as cyber-physical systems, system-of-systems, and digital twins, besides contributing to interoperability and data integration in different knowledge areas.

Future work can proposes a language pattern to OntoUML, the UFO-based UML extension for ontology modeling. In this context, the system ontology could be used to create ontological perspectives to represent the system composition, functional decomposition, system structure, system mechanism, system characterization, and variation over time. Based on the OntoUML for system modeling, a language pattern could also be proposed in ArchiMate to improve the representation of emergence, levels, variation over time, and structural aspects. With this, GST-based modeling notations could be integrated into ArchiMate, such as the Causal Loops diagram [32]. This integration could help better understand the systemic aspects of an organization in a practical way. An important computational application is apply the system ontology to support the use of pattern recognition tecniques to identify emergence patterns from organizational data, specially in complex networks models. Another future work could be the *system ontology* extension to include system behavior and variation over type to better understand how system capabilities manifest through events and how they evolve. In this context, the relations among dispositions (and also capabilities), mentioned in this work, could be more detailed. This ontology could help in system capabilities detailing and digital requirements specification in complex contexts, which involve different kinds of systems, such as system-of-systems, cyber-physical systems, or digital twins. In this context, an implementation of the system ontology in OWL would be usefull in semantic web applications in these kind of system, such as semantic annotation and interoperability. Finally, based on the system capability, other future work is to improve the Competence Ontology [14] to represent better the emergence phenomenon. In this case, the ontology could be used to represent the emergence of organizational capabilities from personal competencies. Besides emergence, another possible improvement concerns the representation of competence development over time.

Acknowledgment. This study was supported in part by CNPq (313687/2020-0), FAPES (281/2021, 1022/2022), and the DSYNE INTPART network (Research Council of Norway project number 309404).

References

1. Ackoff, R.L.: Towards a system of systems concepts. Manage. Sci. **17**(11), 661–671 (1971)
2. Aquino, N.M.R.: A smart assessment of business processes for enterprises decision support. Ph.D. thesis, Université de Lorraine (2021)
3. Azevedo, C.L.B., et al.: An ontology-based well-founded proposal for modeling resources and capabilities in ArchiMate. In: 17th IEEE International EDOC Conference (EDOC 2013), pp. 39–48. IEEE Computer Society Press (2013)
4. Bäcklander, G.: Doing complexity leadership theory: how agile coaches at spotify practise enabling leadership. Creat. Innov. Manag. **28**(1), 42–60 (2019)
5. Baek, Y.M., et al.: A meta-model for representing system-of-systems ontologies. In: Drira, K., Oquendo, F. (eds.) 6th IEEE/ACM International Workshop on Software Engineering for Systems-of-Systems, SESoS@ICSE 2018, Gothenburg, Sweden, 29 May 2018, pp. 1–7. ACM (2018)
6. Bakirtzis, G., et al.: An ontological metamodel for cyber-physical system safety, security, and resilience coengineering. Softw. Syst. Model. **21**(1), 113–137 (2021)
7. Barton, A., et al.: A taxonomy of disposition-parthood. In: Workshop on Foundational Ontology in Joint Ontology Workshops: JOWO 2017, vol. 2050, pp. 1–10. CEUR-WS: Workshop Proceedings (2017)
8. Bernus, P.: Enterprise models for enterprise architecture and ISO9000:2000. Annu. Rev. Control. **27**(2), 211–220 (2003)
9. von Bertalanffy, L.: General System Theory. George Braziller, New York (1968)
10. Bunge, M.: Treatise on Basic Philosophy: The Furniture of the World. Ontology I. Reidel Pub. (1977)
11. Bunge, M.: Treatise on Basic Philosophy. Ontology II: A World of Systems. Springer, Dordrecht (1979). https://doi.org/10.1007/978-94-009-9392-1
12. Bunge, M.: Emergence and Convergence: Qualitative Novelty and the Unity of Knowledge. University of Toronto Press, Toronto (2003)
13. Calhau, R., et al.: Competences in ontology-based enterprise architecture modeling: zooming in and out. Int. J. Softw. Syst. Model. (2023, under review). https://vixra.org/abs/2306.0052

14. Calhau, R.F., et al.: Towards ontology-based competence modeling in enterprise architecture. In: 2021 IEEE 25th International Enterprise Distributed Object Computing Conference (EDOC). IEEE (2021)
15. Compagno, F., et al.: Towards a formal ontology of engineering functions, behaviours, and capabilities. Semant. Web J. (forthcoming)
16. Dori, D., et al.: What is a system? An ontological framework. Syst. Eng. **20**(3), 207–219 (2017)
17. Emery, F.: Socio-technical systems. Systems Thinking (1969)
18. Fine, K.: Things and their parts. Midwest Stud. Philos. **23**, 61–74 (1999)
19. Galton, A., et al.: Dispositions and the infectious disease ontology. In: Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010), vol. 209, p. 400. IOS Press (2010)
20. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. No. 15 in Telematica Institute Fundamental Research Series, Telematica Instituut, Enschede, The Netherlands (2005)
21. Juarrero, A.: Dynamics in action: intentional behavior as a complex system. Emergence **2**(2), 24–57 (2000)
22. Leal, G., et al.: An ontology for interoperability assessment: a systemic approach. J. Ind. Inf. Integr. **16**, 100100 (2019)
23. Lukyanenko, R., et al.: Foundations of information technology based on bunge's systemist philosophy of reality. Softw. Syst. Model. **20**(4), 921–938 (2021)
24. Merrell, E., et al.: Capabilities (2022)
25. Morbach, J., et al.: Onto CAPE 2.0—a (re-)usable ontology for computer-aided process engineering. In: Computer Aided Chemical Engineering, pp. 991–996. Elsevier (2008)
26. Mossio, M., et al.: Emergence, closure and inter-level causation in biological systems. Erkenntnis **78**(S2), 153–178 (2013)
27. Naudet, Y., et al.: Towards a systemic formalisation of interoperability. Comput. Ind. **61**(2), 176–185 (2010)
28. O'Connor, T.: Emergent properties. Am. Philos. Q. **31**(2), 91–104 (1994)
29. Simons, P.M.: Parts: a study in ontology (1987)
30. Skyttner, L.: General systems theory: ideas & applications. World Scientific (2001)
31. Spencer-Smith, R.: Reductionism and emergent properties. In: Proceedings of the Aristotelian Society, pp. 113–129. JSTOR (1995)
32. Sterman, J.: Business dynamics. Irwin/McGraw-Hill c2000 (2010)
33. Wand, Y., et al.: On the deep structure of information systems. Inf. Syst. J. **5**(3), 203–223 (1995)
34. Wegmann, A.: On the systemic enterprise architecture methodology (seam), pp. 483–490 (2003)
35. Yilma, B.A., et al.: Systemic formalisation of cyber-physical-social system (CPSS): a systematic literature review. Comput. Ind. **129**, 103458 (2021)

What Do I Get from Modeling?

An Empirical Study on Using Structural Conceptual Models

Isadora Valle Sousa[1,2](✉), Tiago Prince Sales[2], Eduardo Guerra[1], Luiz Olavo Bonino da Silva Santos[2], and Giancarlo Guizzardi[2]

[1] Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy
{ivallesousa,eduardo.guerra}@unibz.it

[2] Semantics, Cybersecurity & Services (SCS), University of Twente, Enschede, The Netherlands
{t.princesales,l.o.boninodasilvasantos,g.guizzardi}@utwente.nl

Abstract. In the context of enterprises, a wide range of models is developed and used for diverse purposes. Due to the investments involved in modeling, these models should ideally be used in projects in which their benefits outweigh their costs. The analysis of modeling benefits and costs requires an in-depth understanding of the goals of modeling and the properties of models that influence their achievement. This is an issue that has not been sufficiently investigated in the literature. Therefore, we conducted an empirical study to identify and understand the goals modelers aim to achieve through their models, the properties of the models that can aid in the achievement of these goals, and how they assess this achievement. In this study, we focus on a subset of these models, namely structural conceptual models. We found empirical evidence to state that modelers usually achieve more than one goal that can vary among six types of functional goals of modeling and four types of quality goals of modeling. Moreover, according to them, there are six properties of structural conceptual models that can aid in satisfying these goals. Finally, the analysis presented insights into why modelers only subjectively assess the satisfaction of their modeling goals.

Keywords: Conceptual modeling · Modeling goals · Value in modeling

1 Introduction

Models are abstractions of certain aspects of the world that are created and used to meet diverse purposes. Within enterprises, they support complex human activities such as decision-making, training, communication, and systems development. Among the different types of models used within organizations, conceptual models are those focused on defining business entities and their relationships [20]. As such, they can explicitly capture descriptive, prescriptive, and creative aspects of the modeled domain, providing an explicit representation of reality given a level of abstraction and a perspective from which to observe [10,16]. In this study, we focus on a subset of these models, namely structural conceptual

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 21–38, 2024.
https://doi.org/10.1007/978-3-031-46587-1_2

models (also called domain models) [8]. Those are conceptual models that focus on structural (as opposed to dynamic) aspects of the domain aiming at identifying, analyzing, and describing key structural regularities (i.e., types, attributes, relations and constraints) of a specific universe of discourse [9]. Examples of languages that can be used to create such models are Entity Relationship (ER), UML Class Diagram, OntoUML [8], and Object-Role modeling (ORM).

Despite its benefits, modeling activities, such as the development, management, and use of models, require investments in terms of time, money, cognitive effort, etc [10]. Thus, a deeper understanding of the benefits and costs associated with modeling can help modelers to (i) motivate the adoption of a modeling technique in a project or an organization, (ii) convince sponsors to invest in modeling initiatives and teams, and (iii) persuade modeling skeptics to partake in modeling activities. It can also help researchers in developing and improving modeling languages, methods, and tools.

In this paper, we delve deeper into the benefits of modeling—a topic that has not been sufficiently investigated empirically in the literature yet. Assuming that benefits emerge from goal satisfaction [18], we start from Guizzardi and Proper's taxonomy of the functional goals of modeling[1] [10] and further investigate this phenomenon via an empirical study driven by the following research questions:

- RQ1: *Is the taxonomy of modeling goals proposed by Guizzardi and Proper* [10] *sufficient to describe the goals achieved through structural conceptual modeling?*
- RQ2: *What properties of structural conceptual models contribute to the satisfaction of these goals?*
- RQ3: *How do modelers assess the satisfaction of these goals?*

To answer these questions, we conducted nine semi-structured interviews [1] with researchers and practitioners who worked on projects in Brazil, Italy, and the Netherlands. We then performed an inductive and deductive thematic analysis [3,4] on the interview transcriptions.

Our study found evidence that, when creating and using structural conceptual models in practice, modelers are driven by five of the six functional goals proposed by Guizzardi and Proper's taxonomy [10]. We did, however, discover a functional goal not foreseen by them and four recurrent quality goals that are orthogonal to their taxonomy. Additionally, we identified six properties of structural conceptual models that contribute to the accomplishment of these goals, as well as that modelers only subjectively assess the satisfaction of their goals.

The remainder of the paper unfolds as follows. Section 2 explains Guizzardi and Proper's taxonomy. Section 3 discusses the process we followed and the tools we used to collect and analyze the data. Section 4 presents and discusses the results of our analysis of modeling goals. Section 5 positions our work in relation to the state of the art. Finally, Sect. 6 makes some final considerations and discusses the direction of future work.

[1] In [10], Guizzardi and Proper refer to them simply as modeling goals. The distinction between functional and quality goals is well-known in the requirements engineering literature [11].

2 Research Baseline: A Taxonomy of Modeling Goals

Guizzardi and Proper [10] proposed a taxonomy of modeling goals (Fig. 1) based on the notion of *direction of fit* [19] to reason about the goals one may achieve by modeling, the resulting model, or both. For them, "*this notion is meant to connect the propositional content of intentional aspects (i.e., mental states or speech acts) to the external state of affairs of which they are about*" [10]. Guizzardi and Proper argue that models are complex speech acts, so they propose three categories to classify them that are analogous to the ones stated in the notion of direction of fit—World-to-Model (Prescriptive Models), Model-to-World (Descriptive Models), and World-to-Model-to-World (Creative Models).

Prescriptive models are used to intervene in the world. That is, they are used as instruments through which someone brings about changes in the world. Examples include a design that will be implemented, such as a blueprint for a house, and a plan to be followed, such as a BPM model of a process To-Be.

Creative models are those whose existence or recognition in a given community brings about a change in the world. For example, a diagram in a patent file establishes intellectual property rights; a model in a contract outlining the division of a property specifies the ownership rights of each landowner.

Descriptive models are representations of the relationship between conceptualizations (the meaning assigned to a symbol) in the mind of an agent and some existing external reality. In this sense, a model is an instrument through which modelers can represent an individual or collective abstraction of an existing or desired world, such as the blueprint of an actual house or the BPM model of a process As-Is. The creation, manipulation, and communication of this type of model aids in achieving goals related to domain understanding, problem-solving, and domain communication, respectively.

The goal of understanding can be achieved mainly via conceptualization; that is, via the creation of an abstraction in the mind(s) of the modeler(s). This creation process allows both an individual understanding of the domain and its concepts, as well as a collective negotiation of meanings and the formation of a

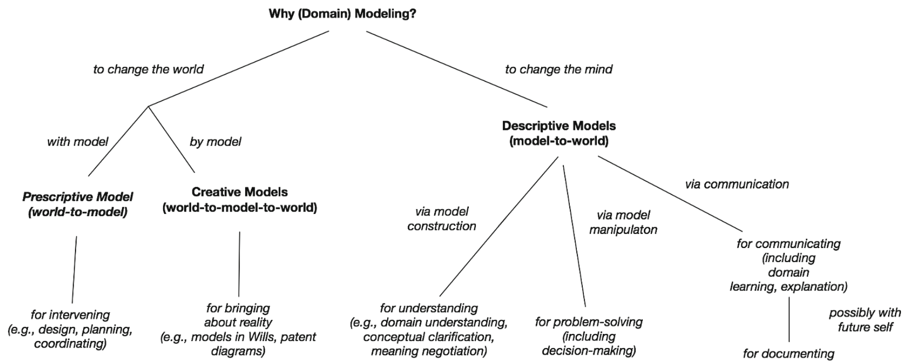


Fig. 1. Guizzardi and Proper's Taxonomy of Modeling Goals [10].

shared conceptualization. To better understand this idea, you just need to think about a time in which you have created a model and how the process of defining concepts and establishing relations helped to shape and develop your and your team's knowledge about a given domain.

When the modeler's goal is not only domain understanding but also the sharing of this understanding, the model can be used for communicating. By employing a model to communicate, the modeler can aim for asynchronous and/or synchronous communication. In the former, the goal is to develop a resource that can be used afterward, by both the modeler herself and others, to communicate an interpretation of the original externalized mental model. The latter, on the other hand, is when this communication happens in real-time interaction. For instance, a model is created as part of the documentation of a software (asynchronous communication); then the model is used to explain the software to new employees (synchronous communication).

An externalized mental model can also be used by those aiming to solve problems and make decisions. In this case, descriptive models are manipulated to support the formation of new beliefs about the world. It happens, for example, when one uses a subway map to decide on which station to exit or when a negotiator uses a game-theoretical matrix to decide which action to take.

According to Guizzardi and Proper [10], the categories proposed in their taxonomy are not mutually disjoint. Therefore, models can be created and used for more than one purpose. An ER model, for example, can be created to guide the implementation of a database schema (for intervening) and afterward be used as documentation for it (for documenting). When the schema needs to be modified, this ER model can help in understanding the schema (for communicating) and in deciding how to modify it (for decision-making). Similar patterns can be observed with the blueprint of a house and with a process model.

3 Research Method

To address our research questions, we conducted a deductive and inductive thematic analysis [3,4] of semi-structured interviews [1] with modelers.

Interviews. Between November and January 2022, the first author conducted 9 interviews with structural conceptual modelers. The interviews were conducted online through Microsoft Teams and each lasted about 60 min. They were conducted in English, Portuguese, and Italian, and analyzed in the original language. When necessary, the quotes used in the article were translated. The interviews were divided into four sections. The first section focused on the background of the interviewees and their knowledge and experiences with models and modeling languages in general. The second section focused on the description (e.g., goals, context, stakeholders, and constraints) of a specific project in which the interviewee has used structural conceptual models. If more than one project came up, both were explored. The third section focused on model creation and use (e.g., goals, facilitators, obstacles, usability, and reusability) and its influence on the project (e.g., modeling costs and benefits). The final section focused on the interviewees' personal opinions about models and modeling (e.g., associated costs and benefits, model reusability, and return of modeling effort).

Participants. Participants in the study were randomly selected from a list developed with authors of relevant literature in structural conceptual modeling, professional networks, and personal contacts of the authors. While developing this list, we focused on modelers who had experience with structural conceptual modeling and had worked on projects in which this type of model was used to solve organizational problems. The selection was interactive until saturation was reached and no new relevant knowledge was obtained from new participants [21]. The professionals were contacted by e-mail and invited to participate in the research. Of the 22 participants invited, 16 agreed to participate. From the 16 interviews conducted, we proceeded to the analysis of only the 9 in which, in the project given as an example, the modeler used a class diagram conceptual model designed with UML or OntoUML. We narrow our focus to guarantee the quality of the analysis since the inclusion of other modeling designs and languages would generate a very heterogeneous sample making it difficult to generalize the results.

In our sample, modelers were located in or worked on projects based in, Brazil (4), the Netherlands (4), and Italy (1). There was 1 female and 8 males. On average, a modeler in our study has a higher education degree in computer science-related fields, has been developing models for longer than 5 years, knows more than one type of model or modeling language, and has worked on more than 20 projects in which models were developed and/or used in diverse domains. All of them actively participated in the development of the model in the project they chose to talk about.

We interviewed modelers working in universities (4), large companies (3), and small companies (2). They were researchers, technical leaders, consultants, and founders. The interviewees who worked in academia chose to talk about projects carried out in partnership with private and governmental organizations. Thus, in all the examples analyzed, the models were used in real-world situations in the fields of energy (2), security (2), environment (1), government (1), healthcare (1), software development (1), and tourism (1).

Research Ethics and Anonymization. During recruitment, participants were informed about the purpose of the study, the content of the questions, and the affiliation of the interviewer. Participants signed an informed consent form and a privacy information sheet acknowledging their knowledge of the purpose of the study and the data management procedures (collection, analysis, and storage). At the beginning of each interview, the interviewer ensured the purpose of the study and the anonymization of the content, explained the dynamics of the interview, and additionally obtained verbal informed consent. As the interviews were conducted through Teams, they were recorded with the participant's permission. The video recordings were transcribed using the NVivo transcription service and manually anonymized by the interviewer.

To protect the identities of the participants, the entire process from conducting the interview to anonymizing the transcripts was done by the same researcher within 24 h of each interview. All personally identifiable information was deleted and the remaining data was stored in a private repository on UNIBZ's GitLab server, with access limited to the research team. We also edited information when quoting participants to avoid personal identification.

Data Analysis. To analyze the transcriptions we developed deductive and inductive coding. We used deductive coding to identify the functional goals of modeling since this is a more theoretical-driven approach in which the researcher codes based on a preliminary list of codes [3]. In our case, this list was composed of the concepts of Guizzardi and Proper's taxonomy [10]. Inductive coding, on the other hand, was used to further investigate participants' goals of modeling and the properties of models that influence their achievement. In this approach, themes emerge from the data, and codes are signed when concepts become apparent in the data. This means that the researcher codes the data without trying to fit it into a pre-existing coding frame or their own analytic preconceptions [3]. To develop the analyses we used the Nvivo 12 software and followed the thematic synthesis process proposed by Braun and Clarke [3]. A researcher began reading the transcripts and familiarizing herself with the data. Next, she identified and labeled specific segments of text generating initial codes that were then randomly selected for analysis and validation by the research group. This procedure helped align the definition of the codes among the group members. In sequence, she translated the codes into themes, sub-themes, and higher-order themes. Then, she revised the themes, the coded extracts, and the entire data to ensure an alignment between them and generate a thematic map of the analysis. At this stage, we introduce a second researcher to analyze and validate the text segments coded and themes created. When there was no consensus between the two researchers, a third party was introduced into the analysis process to mediate the decision about the exclusion, inclusion, or adaptation of a theme or coded extract. The final themes and subthemes generated from the analysis are listed below and discussed in the following section.

- **Functional goals of modeling**: For intervening, For understanding, For problem-solving, For communicating, For documenting, For learning.
- **Quality goals of modeling**: For minimizing effort, For maximizing functional correctness, For maximizing interoperability, For maximizing analyzability and modifiability
- **Model properties**: Reusability, Correctness, Comprehensibility, Completeness, Confinement, Maintainability.

4 Results and Discussion

4.1 Functional Goals of Modeling

The functional goals of modeling are those that modelers try to accomplish through modeling, the resulting model, or both. In Table 1 we present and define the functional goals used as themes in our analysis. The first five were based on Guizzardi and Proper's work [10] and the last one was proposed by us.

Table 1. Functional Goals of Modeling

Goal	Definition	Reference
For Intervening	A goal of using a model as an instrument through which one changes something in the world	[10]
For Understanding	A goal of creating a model to support domain understanding and meaning negotiation	[10]
For Problem-solving	A goal of using a model to guide problem-solving and decision-making	[10]
For Communicating	A goal of using a model to support communication between people about a domain of interest	[10]
For Documenting	A goal of using a model to support *asynchronous* communication between people about a domain of interest	[10]
For Learning	A goal of creating or using a model to learn modeling, modeling methods, and modeling tools	Proposed

The results of our deductive coding showed evidence that five of the six functional goals of modeling proposed by Guizzardi and Proper—"`For Intervening`", "`For Understanding`", "`For Problem-solving`", "`For Communicating`", and "`For Documenting`"—are suitable to describe the goals modelers aim to achieve through structural conceptual modeling (RQ1-F1).[2] However, there was no empirical evidence of modelers' desire to accomplish the goal "`For Bringing About Reality`". We speculate that the reason for that is that the use of models for creating reality due to their sheer existence is indeed much more limited in practice (e.g., patent diagrams, treaties, wills).

The goal of intervening was the most mentioned among modelers, being achieved in ten of the eleven projects given as an example by the interviewees (F1-A).[3] It can be evidenced in the following quote: "*We created a layer that we called the ontological layer. [...] It was a layer to support all the systems. [...] We did the conceptual modeling part, using OntoUML, generated artifacts in OWL and, from then on, we implemented those systems [a semantic search engine, a chatbot, and a legal opinion automator]. We used this conceptual base to develop solutions based on inferences as well.*"

Our analysis showed that the use of structural conceptual models for intervening in the world can be shaped by three factors, the intervention's purpose, type, and result (F1-B). Therefore, the model can be used to change the world by creating or modifying, manually or automatically,

[2] Research Question 1 - Finding 1: It refers to the first find of the study that answers its first research question.

[3] Finding 1-A: It refers to a relevant finding related to Finding 1, but which does not directly answer a research question.

diverse artifacts. For instance, one of the interviewees used an OntoUML model to support the development of an API and any change in the model (e.g. adding a new class) would reflect a change in the API. In this case, the OntoUML model was an instrument through which he could manually create and modify a given artifact. Although no interviewee mentioned doing an automatic intervention, an example would be forward engineering a relational schema from a conceptual model using a method such as that proposed by Guidoni et al. [7].

`For Understanding` was one of the modelers' goals in four projects involving structural conceptual models. One of them said: *"The ontology is an integral part of the standard because it defines what the data is and describes what the interpretation of the information being exchanged is. It provides a starting point for us to understand the nature of the data being exchanged."*

`For Problem-solving` was one of the goals achieved in three projects given as an example by the interviewed modelers. It can be evidenced by the following quote: *"We had discussions with domain experts on the other things in the model, and just by having the model, we could have a better discussion on what should be changed in the data. If you just have a file and you say 'Well, this item in the file needs to be changed', and we do not really understand how this data item relates to other data items. So by having a model, there is more comprehension of the relationships between things."*

Since the term problem-solving is very broad, while defining and codifying this goal, we had to delineate aspects that were not clear in Guizzardi and Proper's work [10]. First, we considered that by "via manipulation" they were also referring to the use and analysis of a model. Second, we noticed that there is a very thin line separating situations in which a model is used for problem-solving and for intervening. In our analysis, we coded the goal `''For Problem-solving''` only in situations in which the model was used to support stakeholders in understanding a given situation and making decisions, but it was not directly used in the intervention process. For example, let us consider again the example of using an ER model to evolve an existing database schema. If one first changes the model and then reflects this change in the schema, we considered it to be an intervention. Conversely, if one used a model simply to identify shortcomings in a schema, we would consider it as problem-solving.

`For Communicating` was one of the goals that drove the interviewed modelers in eight projects. One example can be seen in the following quote: *"So, I think it was fundamental to have an artifact [the model] to guide this communication between two different worlds. They [domain experts] had no knowledge of what ICT [Information and Communication Technology] did, right? That is, they did not have the technical knowledge to know how we actually implemented things, and we also didn't have enough knowledge of the business to be able to develop without support."*

`For Documenting` was the goal achieved in six projects given as an example by the interviewees. One of them said: *"We would want to have, let's say, documentation or knowledge available to have the consistency and the coherence to understand where things live in our organization. Then, it could still be a* *goal to make general models, but not very specific probably, to have this kind of documentation or knowledge management."*

Through the inductive coding conducted in this study, we identified a modeling goal that was not foreseen by Guizzardi and Proper [10]—***"For Learning" (RQ1-F2)***. For Learning was a goal in four projects involving structural conceptual models. The following example was taken from one of the interviews: *"The main goal of the first project was to develop some example models to know if the process of conceptual modeling would be beneficial. So it would be really a test case. We also would do some learning by doing. Then, as a secondary goal, we had also the goal that we, as a team that participated, would also learn OntoUML a bit more."*.

The use of models for learning is an important goal that should be explored more among modelers. Most of our interviewees reported that their models were not used or updated after the project in which they were created ended or after they left the project. They believe this happened because there were no others in the organization/project with the expertise to maintain and use them. For instance, two of them said: *"I think it [the model] should be maintained, but the issue is that I'm leaving the project. So I need someone else to take over my knowledge and there's no one available."* and *"I think they would not be able to follow it on their own because it is a client that does not have the skills to understand this type of model."* They also mentioned difficulties emerging from the lack of knowledge about modeling and modeling language among their project's stakeholders.

Despite these problems, most of the interviewees did not use their models to help others learn. It might be a missed opportunity. By teaching non-modelers to read and understand models, we are allowing them the opportunity to benefit from our models. This can also help to spread the culture of conceptual modeling among those not used to it, bridge the communication gap between modelers and industry, and encourage the continued use, update, and reuse of existing models. One of the interviewees who used the structural models developed in a project for learning purposes was emphatic in highlighting its benefits. He said: *"This involvement, this approach of the client to the conceptual modeling area, I think was so positive that they not only learn, but they develop."*

4.2 Quality Goals of Modeling

The quality goals of modeling are those that determine how the modeling process and/or the resulting model might support the achievement of specific conditions and capabilities in the pursuit or result of a functional goal of modeling. In Table 2 we present and define the quality goals identified in our analysis. After identifying them, we use the ISO/IEC 25010:2011 [13] to support their understanding and definition.

The inductive coding conducted in this study showed empirical evidence of the existence of four quality goals of modeling—"For Minimizing Effort", "For Maximizing Functional Correctness", "For

Maximizing Interoperability", and "For Maximizing Analyzability & Modifiability" (RQ1-F3).

For Minimizing Effort is an objective that goes beyond the desire to communicate or intervene, for example, it is about how these things should be done (e.g. easier, faster, or cheaper). It was one of the goals achieved in five projects given as an example by the interviewees. In the following quote one of them explains how he used the model to simplify and accelerate the development of the artifact in one of his projects: *"Generating a computational artifact without any conceptual modeling reference, going straight to the final product, I think it would be a huge challenge. [...] If you take the model out as an element, we would have to produce the same artifact that we produced in 3 months. I believe we would produce it in a much longer time."* This is an example of using models to minimize both physical and cognitive efforts.

The following three quality goals of modeling are directly related to how the use of a model to intervene in the world can contribute to achieving specific conditions or characteristics in the system arising from that intervention.

For Maximizing Functional Correctness was one of the goals accomplished by three interviewees and can be evidenced in the following quote: *"We use semantics in order to consolidate how data is being processed because a lot of problems and errors in software originate from data being processed in the wrong way."*

For Maximizing Interoperability was one of the goals accomplished by three interviewees and one of them gave the following example: *"I had to create a model that would be able to, at one hand, serve as a kind of a range between the terms that were used in the video environment and the terms that were used in a textual environment."*

For Maximizing Analyzability & Modifiability was one of the goals achieved by three of the interviewees and can be evinced in the following quote: *"It would help future developers to pick it up, read that [the model-based documentation], and understand that the system was made that way. So, if I need to*

Table 2. Quality Goals of Modeling

Goal	Definition
For Minimizing Effort	A goal of minimizing a user's effort in using a model to change the world or people's minds
For Maximizing Functional Correctness	A goal of maximizing the degree to which a system functions correctly by using a model
For Maximizing Interoperability	A goal of maximizing the degree to which systems, products, or components can exchange information and use the information that has been exchanged by using a model
For Maximizing Analyzability & Modifiability	A goal of maximizing the degree to which it is possible to analyze and modify a system by using a model

maintain such a part, from this diagram I can see that it was a given component that is in a given place. So you can go directly to where the component is."

4.3 Model Properties and Their Relation to Modeling Goals

Through inductive coding, we identified six properties of structural conceptual models that, according to modelers, helped in the achievement of their modeling goals—*`Reusability`*,* `Correctness`*,* `Completeness`*,* `Comprehensibility`*,* `Confinement`*, and* `Maintainability` *(RQ2-F4). To better understand these properties and propose the definitions shown in Table 3, we used Mohagheghi et al.'s work on the quality of UML models used in model-based software development [15] as the basis. We chose this model quality framework because the modelers who participated in our study used structural conceptual models for intervening in the world.

`Reusability` ***was the model's property that most helped modelers in achieving their goals (F4-A)***. Eight of them mentioned this during the interview. Regarding reusability, our analysis points out that it can be planned or unplanned and its benefit is directly related to the reduction of effort and costs in the process of changing the world and people's minds. This is because it is easier, faster, and cheaper to reuse a model than to create a new one from scratch. Note that we also considered reuse when the model is extended; that is when it is used as a basis for the creation of another model.

Despite the advantages associated with the reuse of models, we found that the knowledge about the model and its reuse potential is usually restricted to the modelers (F4-B). This can be explained by the fact that in most of the examples given by the modelers interviewed, they created the model to help develop a solution, kept the model to themselves, and did not show or explain it to others involved in the project.

Table 3. Model properties relevant for achieving modeling goals.

Property	Definition
Reusability	The degree to which a model can be reused in a different context and/or for a different purpose than it originally intended
Correctness	The degree to which a model can properly represent the domain, its elements, and their relations
Comprehensibility	The degree to which a model can be understood by its intended users
Completeness	The degree to which a model has the necessary information to fulfill its purpose
Confinement	The degree to which a model only has the necessary information to fulfill its purpose
Maintainability	The degree to which a model can be changed

Interestingly, the only example in which the model was reused by users who were not the initial modelers was given by the only interviewee who reported concern about teaching his project's stakeholders about modeling and involving them in the creation of the model. According to this participant, the initial project consisted of creating an OntoUML model to support the development of three technical solutions. Therefore, reusability in the project context was planned and helped accelerate the software development. What they did not expect, however, was that a different group from another department in the company would use the same model to train new employees. In both cases, the model reuse in the same context for different goals contributed to reduce effort in interventions and communications.

`Correctness` was mentioned by two interviewees. This is a broad term that needs further explanation. First, we do not include the idea of syntax correctness in our definition because there is not any evidence that it actually helped study participants achieve their modeling goals. Second, our definition is centered on the notion of coverage and precision used by Sales in [17]. Therefore, the correctness of the model is related to how well it allows instantiations intended by the modeler and avoids unintended ones. For example, consider the "married to" relation, which holds between two persons. In a common sense ontology about marriage, if a person can marry herself or only marry someone from another gender, the ontology is not precise and has coverage issues respectively. Thus, it has a low level of correctness.

Although in this sense model's correctness can aid in achieving all modeling goals, there are situations in which it is essential to achieve it. These were the case in the two situations considered in our coding process. In both, the model was essential to achieve and maximize the software's functional correctness. For example, one of the participants talked about a project in which the system should be accurate in deciding when and how much tax the company must pay. He said: *"We were working in a domain of sensitive knowledge, where we can have no error margin. So, to have categorical knowledge, you need to use artifacts derived from a categorical approach."*

`Comprehensibility` was mentioned by six interviewed modelers. This property is the key to reaching all goals related to the use of models since to properly use them you should be able to understand them. However, comprehensibility is not an absolute property; it depends on factors that can affect how well it supports users in achieving a given modeling goal. For instance, the type of domain represented, the modeling language used, the level of abstraction chosen, and the dimension and structure of the diagrams may all have a different effect on the user's comprehension, depending on their modeling expertise as well as the given domain and language. One of the interviewees highlighted the importance of the diagram's layout to model comprehension and acceptance. He said: *"I also learned something else in this project where I was the guy that aligned the diagrams. [...] So, for people not used to reading models, I'm sure that the position, the presentation and the colors and everything, that's really important. And even that important, they will say that they like it very much or don't like it at all."*

`Completeness` and `Confinement` were cited by six and three interviewees respectively. The former increases the quality of users' changes in the world and minds by allowing them to fully represent and understand a given reality. The latter reduce their effort in the processes of making these changes by preventing them from wasting time on unnecessary things. Regarding completeness, one of the interviewees said that he knows that the model is complete when it can answer all of the project's team questions. He said: *"This was an easy project to do because the competency questions were kind of look-up questions. We had these questions that we wanted to answer. So the depth of the model was very much based on that."*. Regarding confinement, another interviewee mentioned that while creating the ontology the modelers only focused on the information they previously defined as necessary. He said: *"We listed what types of information we wanted to exchange, what the agents thought was relevant to exchange, and we only went as far as the ontology could describe them. We didn't introduce anything into the ontology that went beyond describing that information."*

`Maintainability` was pointed out by two participants. According to them, this property can positively affect the functional goal of documenting and the quality goal of maximizing software modifiability. In these cases, the model should be constantly changed to properly represent reality increasing the effectiveness of users when documenting and modifying artifacts. Furthermore, on the one hand, maintainability can affect the model's correctness by facilitating its updating. On the other hand, it can be affected by the model's comprehensibility since the more complex the model is, the more difficult it is to maintain. One of the interviewees talked about that in the following quote: *"Maintenance is a quality that should be an architectural concern when you develop your data. So it will have an impact on our own software and also on your models, since other people must be able to read your model in order to maintain it."*

4.4 Assessing the Satisfaction of Modeling Goals

If models are created and used to satisfy certain goals, assessing their satisfaction seems to be necessary to know whether or not modeling was worthwhile. When analyzing our interviewees' answers on this subject, ***we found out that modelers only subjectively assess the satisfaction of their modeling goals due to the difficulties in identifying and measuring the benefits of the modeling process and/or the resulting model (RQ3-F5)***.

Some of them argue that not all benefits of modeling can be defined at the beginning of the project since it is difficult to foresee all possible uses of the model. Sometimes the model is created for a specific goal and used for other unplanned purposes, such as in the example above in which the model was created to guide the development of systems and later used to train new employees.

There are also reverse situations in which the expected benefits are not achieved due to reasons such as a lack of modeling culture in the company or the modeler's resignation from the project. For example, one of the interviewees reported that he was hired as a consultant to develop a model to mediate the external communication between the company and its clients. However, he knows that the model could also be used for internal communication among the company's modelers. Unfortunately, he doesn't know if this will happen since his job was only to develop the model and the company has no modeling culture.

The interviewed modelers also mentioned the difficulty of measuring the modeling benefits to justify their subjective analysis. According to them, even when they identify these benefits, they don't know how much they contributed to achieving the modeling goal. Indeed, it is harder to measure how much a model facilitated the development of a system, accelerated a decision-making process, or increased the modifiability of software. The following quote from one of the interviewees illustrates this difficulty: *"I don't know if there are techniques capable of measuring, for example, that I reduced something. I have no way of measuring, for example, that I reduced meeting time to make a decision by 50% because I used an ontology [in OntoUML] to agree on what data people wanted to exchange."*

Two respondents argued that an objective analysis of the satisfaction of modeling goals would be worthwhile if one could analyze diverse scenarios to evaluate which option would offer the best cost benefit. Although we agree with this claim, in most modeling initiatives it is not possible to concretely compare alternatives. This does not mean that an objective cost-benefit analysis is not beneficial. It might not help you prove that model-driven software development is better than non-model-driven software development. Still, it might help you demonstrate that the former is good enough to justify the investments in it. This may help convince those skeptical of modeling to adopt it, which is, according to most of the interviewed modelers, a difficult task.

Finally, ***our results showed that due to the difficulties in identifying and measuring the benefits of a modeling initiative, modelers do not analyze them to decide when to create or use structural conceptual models in their projects but to justify them (F5-A).*** Modelers do not seem to want to know when it is worth modeling. They always choose to solve problems through models, even if they do not rationally know whether modeling is the best option for a given situation. They will model because their experience shows that modeling was a good option in similar cases and they use it to base their modeling decisions such as when to model, when to stop modeling, or what modeling language to use. For instance, one of the interviewees said: *"I have a quick discussion, I identify the problem, and I already know the approach we are going to work with, I already know the language. Experience is everything!"*

4.5 Threats to Validity

We believe that our study could be affected by the rosy view phenomenon. This phenomenon is associated with the fact that within days after the event, people have much more positive evaluations of the event [14]. Therefore, by asking the practitioner to choose a project and talk about it, we take the risk of getting a more positive version of their modeling experience. Although this positive view of the facts can not affect the functional goal of modeling, it can affect the quality goal of modeling. To overcome the self-reported data challenges, sufficient rigor and care were applied in covering the subjects through multiple questions and solicitation of examples.

We understand that online interviews may have some disadvantages compared to face-to-face interviews, but the latter was not an option in this study because the participants were geographically dispersed. Although some authors argue that interviewees' performance is usually typically inferior in the online setting [2], this was not a problem for our study since we did not ask the interviewee to perform any task and did not seek to evaluate their performance or behavior throughout the interview. We concentrated on the content of their responses. Moreover, the use of video and the interviewer's expertise effectively mitigated any potential consequences of the online interview on social presence, eye contact, and impression management as perceived by the participants.

The modest size of the sample used in the study can also be seen as a threat to the validity and generalization of the results. Despite its size, we believe that the sample selected faithfully represents the scenarios and population that we intend. The diversity of contexts of the projects given as examples and the high experience in structural conceptual modeling of the modelers interviewed helped us to create a meaningful data set. In addition, the sample size complies with the saturation criterion associated with the chosen research method [21].

In thematic analyses such as the one developed, the way the coding is conducted might increase the risk of bias in the data analysis. To minimize this risk, we coded the complete data twice, one for each coding method (deductive and inductive) separately, and used more than one researcher to analyze and validate the text segments coded and themes created.

Finally, other possible research biases can be discussed. First, we interviewed only professionals who were already modelers and did not interview those who chose not to model. Second, we interviewed only one woman. However, we believe that for this research, they are not relevant.

5 Related Works

Davies et al. [5] conducted a survey in Australia to, among other aims, investigate practitioners' motivation for doing conceptual modeling. They asked respondents to choose among 17 pre-defined goals. The most frequent one ended up being to design and maintain databases. This result is in alignment with our findings about the goal of intervening. The authors investigated factors that might encourage or discourage the use of modeling techniques (e.g. ER, UML). However, their results are more related to the properties of modeling tools than the properties of the models themselves. They did not investigate how modelers assess the achievement of their modeling purposes.

Dobin and Parsons [6] empirically investigated how much and for what purposes UML components (e.g. class and use case diagrams) were used. They found that class diagrams were the most used ones and they were mainly used to support the understanding of the application among technical members of the project team. The authors also investigated the effectiveness of UML components in facilitating communication within software development teams. For this purpose, class diagrams were rated as "moderately useful" by most of the respondents. They did not investigate why or how they facilitate communication and did not extend their analysis to other modeling purposes.

Ho-Quang et al. [12] empirically analyzed the motivation and benefits of the use of UML models within software development teams. Their results showed that collaboration is the most important motivation for using UML. They also claim that the use of UML positively affects the project's planning phase and development process, as well as the communication among team members. This last one helps to attract new contributors and facilitate their integration.

Both Dobin and Parsons [6] and Ho-Quang et al. [12] had similar aims to ours in terms of investigating the goals related to structural conceptual modeling. However, they had a more restricted scope regarding the type of models and goals analyzed and did not mention the use of a theory to support the modeling goal selection employed. Despite these differences, our results align with theirs as they also identified the benefits of using UML class diagrams for intervening, documenting, and communicating in software development projects.

Finally, regarding the assessment of modeling goals, there is one main theoretical work that addresses the notion of Return on Modeling Effort (RoME). In their paper, Proper and Guizzardi [16] deepened the discussion of RoME and introduced the idea of a model's Value in Action (ViA) and the need to manage the Retention of Modeling Effort (RiME). The authors also stressed the demand for a more rigorous underpinning of the costs and benefits involved in domain modeling. This is not only in line with the aim of our study but also highlights its importance. Although they enriched the discussions on modeling value-based assessment, no empirical evidence of their work was presented.

6 Conclusion

The study we reported in this paper focused on identifying and understanding the goals modelers aim to achieve through structural conceptual modeling (RQ1), the properties of models that can aid in the achievement of these goals (RQ2), and how they assess this achievement (RQ3).

Our empirical study found evidence that, when creating and using structural conceptual models in practice, modelers are driven by five of the six functional goals proposed by Guizzardi and Proper's taxonomy ***(RQ1-F1)***. Throughout the study, we clarified and refined the authors' definitions. We also discovered a functional goal not foreseen by them ***(RQ1-F2)*** and four quality goals that are orthogonal to their taxonomy ***(RQ1-F3)***. We also identified six properties of structural conceptual models that can aid in the accomplishment of these goals ***(RQ2-F4)***. Additionally, our analysis uncovered that modelers only subjectively assess the satisfaction of their modeling goals due to the difficulties in doing so***(RQ3-F5)***. The new functional goal (`for learning`) can be seen as a subcategory of the use of models *to change the modeler's mind*, but now about the modeling approach itself as opposed to about the domain.

Our study also found that the goal `''For Intervening''` was the most frequent among projects involving structural conceptual models ***(F1-A)*** and that it can be shaped by three factors, the intervention's purpose, type, and result ***(F1-B)***. Moreover, we found that reusability was the property that most helped modelers in achieving their goals ***(F4-A)***, although we uncovered that the knowledge within the model and its reuse potential is usually restricted to the modelers ***(F4-B)***. Finally, our results showed that modelers do not analyze their modeling goals to decide when to create or use structural conceptual models in their projects but to justify them ***(F5-A)***.

Our findings can serve as a guide to model evaluation. A clear understanding of why and how modeling should be done can help modelers to motivate the adoption of a modeling technique, convince sponsors to invest in modeling initiatives, and persuade modeling skeptics to partake in modeling activities. It can also help researchers in developing and improving modeling ecosystems.

In subsequent work, we plan to investigate the costs of structural conceptual modeling, the barriers and incentives for the adoption of its modeling techniques, and the perspective of those against it.

References

1. Adams, A., Lunt, P., Cairns, P.: A qualitative approach to HCI research (2008)
2. Basch, J.M., et al.: It takes more than a good camera: which factors contribute to differences between face-to-face interviews and videoconference interviews regarding performance ratings and interviewee perceptions? J. Bus. Psychol. **36**, 921–940 (2021)
3. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**(2), 77–101 (2006)
4. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: International Symposium on Empirical Software Engineering and Measurement, pp. 275–284. IEEE (2011)
5. Davies, I., et al.: How do practitioners use conceptual modeling in practice? Data Knowl. Eng. **58**(3), 358–380 (2006)
6. Dobing, B., Parsons, J.: How UML is used. Commun. ACM **49**(5), 109–113 (2006)
7. Guidoni, G.L., et al.: Forward engineering relational schemas and high-level data access from conceptual models. In: Conceptual Modeling, pp. 133–148 (2021)
8. Guizzardi, G.: Ontological foundations for structural conceptual models (2005)
9. Guizzardi, G., Halpin, T.: Ontological foundations for conceptual modelling. Appl. Ontol. **3**(1–2), 1–12 (2008)
10. Guizzardi, G., Proper, H.A.: On understanding the value of domain modeling. In: 15th International Workshop on Value Modelling and Business Ontologies (2021)
11. Guizzardi, R.S., Li, F.L., Borgida, A., Guizzardi, G., Horkoff, J., Mylopoulos, J.: An ontological interpretation of non-functional requirements. In: FOIS, vol. 14, pp. 344–357 (2014)
12. Ho-Quang, T., et al.: Practices and perceptions of UML use in open source projects. In: International Conference on Software Engineering, pp. 203–212 (2017)
13. ISO: Systems and software engineering - systems and software quality requirements and evaluation (SQuaRE) - system and software quality models (2011)
14. Mitchell, T.R., et al.: Temporal adjustments in the evaluation of events: The "rosy view". J. Exp. Soc. Psychol. **33**(4), 421–448 (1997)

15. Mohagheghi, P., et al.: Definitions and approaches to model quality in model-based software development-a review of literature. Inf. Softw. Technol. **51**(12), 1646–1669 (2009)
16. Proper, H.A., Guizzardi, G.: Modeling for enterprises; let's go to RoME ViA RiME. In: The Practice of Enterprise Modeling 2022 Forum, vol. 3327 (2022)
17. Sales, T.P.: Ontology validation for managers. Master's thesis, Federal University of Espírito Santo (2014)
18. Sales, T.P., Baião, F., Guizzardi, G., Almeida, J.P.A., Guarino, N., Mylopoulos, J.: The common ontology of value and risk. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 121–135. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_11
19. Searle, J.R.: Intentionality: An Essay in the Philosophy of Mind. Cambridge University Press, Cambridge (1983)
20. Snoeck, M.: Enterprise information systems engineering: the merode approach (2014)
21. Tong, A., Sainsbury, P., Craig, J.: Consolidated criteria for reporting qualitative research (COREQ): a 32-item checklist for interviews and focus groups. Int. J. Qual. Health Care **19**(6), 349–357 (2007)

A Generic and Customizable Genetic Algorithms-Based Conceptual Model Modularization Framework

Syed Juned Ali(✉), Jan Michael Laranjo, and Dominik Bork

TU Wien, Business Informatics Group, Vienna, Austria
{syed.juned.ali,dominik.bork}@tuwien.ac.at

Abstract. Conceptual models need to be comprehensible and maintainable by humans to exploit their full value in faithfully representing a subject domain. Modularization, i.e. breaking down the monolithic model into smaller, comprehensible chunks has proven very valuable to maintain this value even for very large models. The quality of modularization however often depends on application-specific requirements, the domain, and the modeling language. A well-defined generic modularizing framework applicable to different modeling languages and requirements is lacking. In this paper, we present a customizable and generic multi-objective conceptual models modularization framework. The multi-objective aspect supports addressing heterogeneous requirements while the framework's genericity supports modularization for arbitrary modeling languages and its customizability is provided by adopting the modularization configuration up to the level of using user-defined heuristics. Our approach applies genetic algorithms to search for a set of optimal solutions. In this paper, we present the details of our Generic Genetic Modularization Framework with a case study to show *i*) the feasibility of our approach by modularizing models from multiple modeling languages, *ii*) the customizability by using different objectives for the modularization quality, and, finally, *iii*) a comparative performance evaluation of our approach on a dataset of ER and ECore models.

Keywords: Modularization · Genetic Algorithm · Generic Framework

1 Introduction

Conceptual modeling enables domain understanding and communication among stakeholders and is integral to enterprise and information systems engineering [30]. The usefulness of any conceptual model for humans is inversely proportional to the size of the model depicted. Models with more than 30 nodes/edges are considered to be already challenging for easy comprehension. The more relationships in a model, the less comprehension is possible due to the accompanying increase in complexity [41]. Therefore, the increased size and complexity can make models *cognitively intractable* [12].

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 39–57, 2024.
https://doi.org/10.1007/978-3-031-46587-1_3

Clustering or modularizing conceptual models[1] into smaller chunks provides benefits in that very large models become easy to communicate, validate, and maintain [41]. However, to be truly effective in improving human understanding, clustering approaches must be based on sound principles of human information processing [25].

Due to the NP-hardness of the modularization problem, search-based algorithms treat modularization as an optimization problem over an objective function [28] using techniques like genetic algorithms (GA) [9,26]. However, several limitations exist, such as choosing the heuristic for modularization cannot be generalized well given the diversity of conceptual models designed, domains modeled, modeling languages used, and requirements being addressed. Finally, several steps are involved in GA-based solutions i.e., selection, crossover, and mutation which should be customizable depending on the modularization case. Therefore, there is a need for a modularization framework that *i*) can be adapted to multiple modeling languages, *ii*) provides multiple heuristics and allows extending the set of heuristics; and *iii*) allows GA parameter (re-)configuration based on the modeler's requirements.

In order to achieve modularization genericity in terms of modeling languages, we need a generic representation of conceptual models that can be used as input for the modularization techniques. Knowledge graphs (KGs) can encapsulate knowledge in a graph structure, creating new processing possibilities, such as knowledge reasoning. KGs provide a foundation for data integration, fusion, analytics, and sharing [34] based on linked data and semantic metadata. Conceptual models can be treated as graphs with nodes and edges capturing conceptual model-specific information. We can transform a general conceptual model into KGs and utilize the benefits of a KG-based representation of conceptual models. Therefore, we can use knowledge graphs as a generic representation of conceptual models that captures the domain semantics of the model and the model's graph structure.

Once we have a generic conceptual model representation, we need to drive the modularization toward optimization objectives suitable for the modeling requirements. Multiple objectives can measure the modularization quality. Moody and Flitman [25] proposed nine principles for decomposing data models, including the number and the size of cognitively manageable clusters, which are also valid for conceptual models. Moreover, users should also be able to define objectives based on their requirements. E.g., a custom objective to optimize the modularization for the number of abstract classes in a module. Furthermore, using an objective as a combination of several objectives can be beneficial.

To that end, we present the main contributions of this paper as follows - *i*) we introduce the Generic Genetic Modularization Framework (GGMF) for conceptual models that provides a multi-objective, generic, customizable framework for modularizing conceptual models; *ii*) we provide a UI platform for modeling experts to modularize conceptual models where users can upload a con-

[1] Note, we use graph clustering/partitioning and modularization interchangeably in this paper.

ceptual model, define configuration parameters for the GA-based optimization, and select the objectives for which the GA should optimize. Furthermore, the user can weigh each objective based on its expected importance. Therefore, the user can execute the optimization as a multi-objective optimization or aggregate the objective into a single objective in a weighted or unweighted manner. Finally, (*iii*), we present a comparative analysis of our approach with existing approaches. We evaluate our framework based on the following research questions -

RQ1- Is developing a modularization framework that supports multiple modeling languages feasible?
RQ2- How does the framework support modularization for different requirements?
RQ3- How does the modularization perform compared to existing approaches?

Note that the focus of our paper is the genericity and customizability provided by GGMF. Therefore, we do not focus on the details of the optimization mechanism of genetic algorithms and use the jenetics [2] library. The rest of the paper is structured as follows: Sect. 2 presents the foundational concepts involved in our work. Section 3 discusses the relevant literature while Sect. 4 introduces our framework and the developed tool support. Section 5 evaluates our approach. We discuss the threats to validity in Sect. 6 and finally, we conclude this paper in Sect. 7.

2 Foundations

2.1 Modularization

Modularization in conceptual modeling concerns the decomposition of a monolith, potentially overarching model, into smaller, more comprehensive model chunks—called modules. The module components depend on the intended purpose of modularization while fulfilling the definition of a module. E.g., if the purpose of the module is solely to answer a query, then it should only be composed of the necessary concepts and relations that can answer the considered query [21].

2.2 Conceptual Knowledge Graphs

Knowledge Graphs (KGs) represent a collection of interlinked descriptions of entities—e.g., objects, events, and concepts. KGs provide a foundation for data integration, fusion, analytics, and sharing [34] based on linked data and semantic metadata. Recently, a generic approach has been proposed to transform arbitrary CMs into CKGs called *CM2KG* [36]. However, CM2KG focuses only on the element labels and metamodel information. Ali et al. [3] define the notion of Conceptual Knowledge Graphs (CKGs) by adding semantics from external sources such as language metamodel, domain, and foundational ontologies to the KG-based representation. These enriched semantics can be used as information sources for many tasks. In our work, CKGs act as the intermediary representation of CMs based on which the modularization is performed.

2.3 Genetic Algorithms

Genetic algorithms (GA) are randomized search-based optimization algorithms inspired by the principles and mechanics of natural selection and natural genetics [15]. GAs map the optimization problem into the concepts involved in GA-based search and apply the search to find the optimal (a set of) solutions. We briefly describe the involved concepts.

Genetic Encoding. The genetic encoding i.e., the *genotype* is the representation of the optimization problem that the GA uses as input for executing the optimization. In modularization, a *genotype* could be a binary string or an array where each element represents the module assignment of a node in the graph. A *gene* is a specific element within a *genotype*, e.g., a gene would represent the module assignment of a specific node in the graph. The value stored in a gene is called an *allele*. A *chromosome* is a collection of genes that forms a complete genetic representation. It represents an individual or candidate solution in the genetic algorithm, e.g., a complete set of module assignments for all nodes in the graph. Commonly used encoding schemes are binary encodings, tree encodings, or matrix encodings [19]. A *population* is a set of currently present chromosomes. Linear Linkage Encoding (LLE) is a genetic encoding for grouping problems, such as graph coloring or data clustering. It represents a problem solution as a fixed-length chromosome, which encodes the assignment of elements to a group. This encoding ensures there is only one representation for a grouping solution [20], thereby mitigating redundancy and isomorphism.

Fitness Functions and Objectives. The fitness function is used to determine the fitness value of a chromosome [15]. The fitness value quantifies the quality of chromosome w.r.t a fitness function. In the context of modularization, the fitness function would evaluate the quality of a chromosome by considering factors such as intra-module similarity and inter-module dissimilarity. GA can use multiple objectives during optimization. However, multi-objective problems often have conflicting objectives, i.e., maximizing one objective may lead to minimizing another, e.g., maximizing cohesion and minimizing coupling. Pareto optimal solution sets or Pareto front is a solution for conflicting objectives. This set consists of possible Pareto optimal solutions.

We explain modularity and MQ in Eq. 1.

$$\begin{aligned} modularity &= \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j) \\ MQ &= \sum CF_i \text{ where} \\ CF_i &= \frac{2\mu_i}{2\mu_i + \epsilon_i} \end{aligned} \tag{1}$$

For a graph G = (V,E) where V is the set of nodes in the graph and E is the set of edges. A is the adjacency matrix of a graph with n nodes and m edges. Modularity is defined in Eq. 1, where A_{ij} is an element of the adjacency matrix A, $\frac{d_i d_j}{2m}$ is the expected number of edges between nodes i and j, d_i is the degree of node i, c_i is the community to which node i belongs and δ is the Kronecker delta function that returns 1 if $c_i = c_j$ and 0 otherwise. The MQ score is obtained from the sum of all module factors. CF_i denotes the module factor for each module i, where μ_i represents the count of intra-module edges and ϵ_i represents inter-cluster between two modules [28].

Selection. Once we have a genetic encoding, the selection step selects a subset of chromosomes from the current population for reproduction based on their quality defined by a fitness function and creating the offspring [19]. The fitness function guides the selection process by favoring individuals with higher fitness scores, increasing the probability of their genetic material being passed on to the next generation. The selected individuals for the reproduction process are also called the mating pool. There are different types of selection operators in GA. The most common variations for single-objective GA are roulette wheel section, rank selection, and tournament selection [19]. The roulette wheel selects chromosomes randomly for the reproduction process [15] based on the number of copies of each chromosome present in the population. Tournament selector uses a tournament process, where randomly selected individuals compete against each other in pairs or larger tournaments. The winner is the individual with the highest fitness, and the winner is added to the mating pool with chromosomes with higher average fitness compared to the population's average fitness. The tournament is repeated until the desired size for the mates is reached.

Alterers. *Crossover* combines the genetic material from two parent chromosomes to create the offspring. In the context of modularization, the crossover could involve combining the module assignments of two parent chromosomes to generate a new set of module assignments for the offspring. *Mutation* is an operator that introduces chromosome alterations. It helps to introduce diversity in the population and explore new solutions. In modularization, the mutation could involve randomly reassigning the module of a node to a different cluster.

3 Related Works

We now discuss the different modularization approaches, separating GA and non-GA approaches, and the modularization metrics proposed in the literature.

3.1 Conceptual Models Modularization

Non-GA-Based Approaches. Stuckenschmidt and Klein [37] propose a method that clusters models based on the structure of the class hierarchy for real-world ontologies like SUMO and the NCI cancer ontology. Saruladha et al. [33] propose two new neighbor-based structural proximity measures, TNSP and DNSP to decompose ontologies into disjoint clusters. They consider concept pairs with common neighbors for clustering. Doran et al. [11] present a language-independent ontology module extraction approach implemented as ModTool. Andritsos et al. [4] present LIMBO, a hierarchical clustering algorithm based on minimizing information loss that will be incurred on merging two nodes in a cluster, in the context of a software system. In [24] a hierarchical clustering-based weighted linkage clustering (WLC) approach is presented. They associate a new feature vector using the feature vectors of a set of cluster entities. Two entities are merged based on their types, globals, and routines. Hence, the new feature vector correctly reflects relationships between the entities. Pourasghar et al. [28] present a GMA (Graph-based Modularization Algorithm) modularization technique. Their work uses relationship depth to compute the similarity between model entities. Furthermore, they propose several metrics to evaluate the modularization quality that uses structural features of the modularized model.

Metrics. Sarkar et al. [32] propose a set of metrics for determining the quality of modularization of large-scale object-oriented software using intra-module dependencies, the modules' APIs, and object-oriented inter-module dependencies, e.g., inheritance. Using model slicing, Bae et al. [5] modularize UML metamodels. The Model Slicing approach uses vital elements to generate the modularized metamodel. A key element is a model element of M. The slicing is executed in two phases using the edges of the multigraph and the key elements to determine the slices. Hinkel et al. [16] apply the modularization quality metrics proposed by [32] on metamodels. Like class diagrams, metamodels can be organized into packages, making the metrics appealing for application on metamodels. Based on the results of this work, they propose an entropy-based approach. Their metric measure the degree of classes that are stored in different packages [17].

GA-Based Approaches. Bork et al. [9] introduce the ModulER tool for modularizing entity-relationship models. The tool follows a meta-heuristic search approach using genetic algorithms. Multiple objectives, defined as fitness functions, aim to minimize or maximize specific properties of the modularization of each individual, resulting in a Pareto Set of optimal solutions [9].

Mu et al. [26] propose a hybrid genetic algorithm (HGA) using a heuristic based on edge contraction and vectorization techniques to generate feature-rich solutions and subsequently implant these solutions as seeds into the initial population. Finally, a customized genetic algorithm (GA) improves the solution quality. Tabrizi et al. [38] combine hierarchical clustering with genetic algorithms, where they first modularize the model using GA and then further improve the solution using hierarchical clustering. Bavota et al. [6] propose Interactive Genetic Algorithms (IGAs) to integrate the developer's knowledge in a re-modularization task. Their approach uses automatically evaluated fitness functions and a human evaluation to penalize cases where a developer considers module assignments meaningless.

3.2 Modularization Metrics

Metrics are needed to quantify the modularization quality and different metrics have been proposed in the past. Some metrics use the characteristics between modules and relationships between elements in modules. Others utilize concepts from information theory or network theory. Lastly, some metrics consider specific conceptual models' properties and are only tailored for a specific conceptual modeling language.

Moody et al. [25] propose nine principles for decomposing data models using network theory and cognitive science principles. Sarkar et al. [32] present metrics for cohesion and coupling between modules by defining an entropy metric that measures the extent to which classes are used together and should be clustered. The properties or structure of specific conceptual models can also indicate which elements belong together in a module. Prajapati et al. [29] modularize by properly distributing classes among various packages in a model with minimum perturbation. Hinkel et al. [16] adapt some of the proposed metrics in the context of metamodels. Hinkel et al. [17] propose an entropy-based modularization metric that quantifies class distribution in packages [17]. Dazhou et al. [18] use weights for relations in UML classes. Each UML class relation gets weight assigned to build weighted class dependence graphs. Singh et al. [35] similarly assign dependency weights to BPMN models.

In network theory, many metrics can be used to measure modularity. The importance of an edge is captured by determining the shortest paths of all vertex pairs that go through the edge. The edges have higher values in communities as there are more shortest paths between the vertex pairs. This metric is called Edge Betweenness Centrality [13]. Similarly, Vragovic et al. [42] use the idea that neighbors in communities are close to each other, even when they are removed. They introduced the concept of loop coefficients. It takes the number of smallest loops running through a node into consideration. High coefficients indicate main nodes in a community, whereas low values indicate peripheral nodes in communities. Newman [27] defines a spectral method for modularity optimization. His approach uses Eigenvectors to express modularity characteristics and optimize accordingly.

In the related work, we note a need for a customizable and generic modularization framework, i.e., applicable to multiple modeling languages, extensible to user-defined metrics. Multiple modularization metrics available can be used depending on the requirements. With this motivation, we propose our generic and customizable conceptual model modularization framework using genetic algorithms.

4 The Generic Genetic Modularization Framework (GGMF)

In the following, we introduce our proposed Generic Genetic Modularization Framework (GGMF) in detail. We show in Fig. 1 the end-to-end approach from a monolithic conceptual model to a modularized one. The conceptual model is first transformed into a CKG, which the GA then uses to produce a Pareto Set of optimal solutions based on the user's selected objectives and requirements.

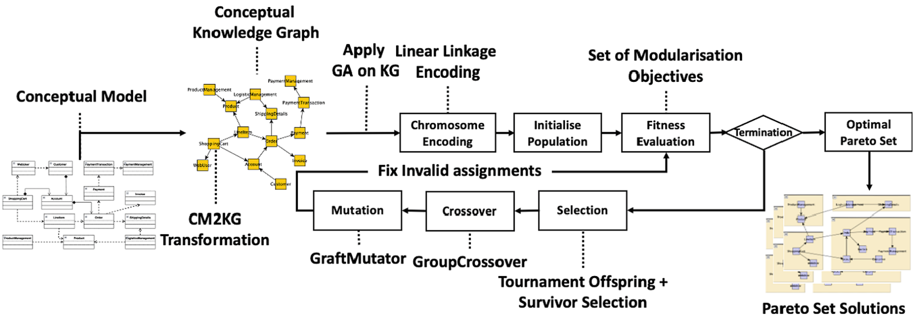


Fig. 1. End-to-end Generic Genetic Modularization Framework

4.1 Model Transformation and Genetic Encoding

In GGMF, we first transform a conceptual model of any modeling language into a CKG. We show in Fig. 2 the transformation of an Online Shop UML class diagram into the corresponding CKG. The CKG is represented as a labeled property graph [14] with each node and edge capturing their corresponding properties, e.g., association, and dependency relationships. The GA can use these properties during optimization (see Fig. 4).

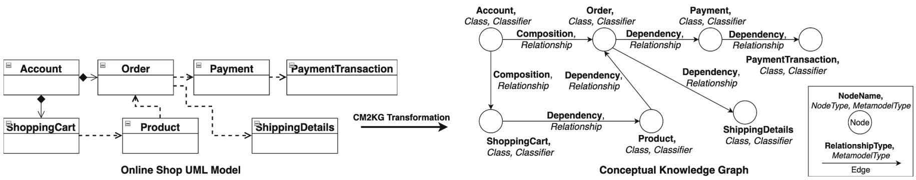


Fig. 2. Conceptual Model to CKG transformation

Once we have transformed the model into a CKG, we create a genetic encoding that serves as input to the GA. We use Linear Linkage Encoding of the CKG in Fig. 3. LLE represents the elements in each module as a linked list with elements sorted in the order of their indices. The sorting is done to avoid isomorphic representations of a module and thereby avoid duplicate solutions. Figure 3 shows LLE as a chromosome representing the CKG where each vertex and edge is assigned to a module. A gene is associated with each vertex and an edge. The allele values of each gene are the index of the next vertex or edge in the same module. So each gene is linked to the next element in the module except for the last element. The last element points to itself. Each module is denoted by the last element, i.e., the ending node. The modularization yields a set of modules where each module is connected to the other. Note that in the subsequent subsections, we provide high-level details of the aspects involved in the optimization process of GAs. These aspects are invariants during optimization in GGMF and do not contribute to the genericity and customizability of GGMF; therefore, the in-depth details about these aspects are outside the scope of our paper.

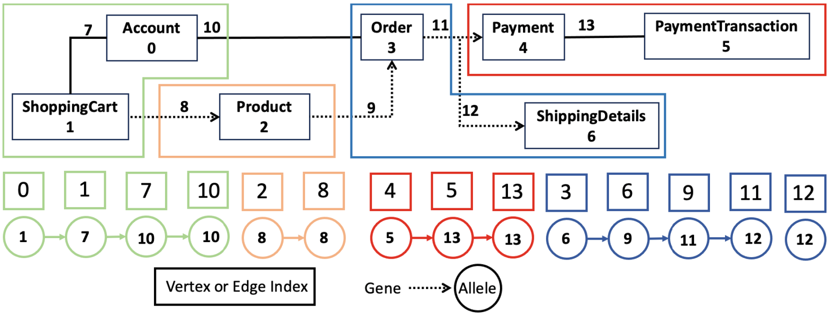


Fig. 3. Linear Linkage Encoding Example

4.2 Objective Functions

Given an LLE chromosome, we can evaluate the module assignment of each element of the CKG and therefore evaluate the modularization quality on the objectives selected for modularization. Table 1 shows multiple objectives to measure modularization quality from a chromosome found in the literature. We categorized the objectives into *module-based*, *semantics-based*, *entropy-based*, and *graph-based*. The module-focused objectives aim to measure the module-related properties. The semantics-based objectives use language model-based word embeddings of an element. The entropy-based types only apply entropy techniques to the string values. Lastly, graph-based objectives employ graph properties for characterization. The categorization is not mutually exclusive, i.e., one objective has side effects on other objectives. For example, graph-based objectives can capture cohesion.

We can use the structural features of the graph to evaluate the module and graph-based objectives. Note that module and graph-based objectives can have edge weights depending on the relation type (see Fig. 4). To evaluate the entropy and semantics-based objectives, we use the natural language semantics captured by the node labels using a language model-based representation of the nodes that capture the natural language semantics of its labels. Recent contextualized NLP

Table 1. List of implemented objective functions

Category	Objective	Objective Description
Module	Cohesion	Maximise the sum of intra-module edges
Module	Coupling	Minimise the sum of inter-module edges
Module	Balancedness	Minimize the standard deviation of module size from a threshold[a]
Module	Average Module Size	Minimise the average module size
Semantics	Semantic cohesion	Maximise the semantic similarity of intra-module vertices
Semantics	Semantic coupling	Minimise the similarity of inter-module vertices
Entropy	String similarity	Maximise the average string similarity per module
Entropy	String difference	Maximise the average string difference between modules
Graph	node closeness	Minimise the average of vertex closeness centrality per module
Graph	edge betweenness	Minimise the average edge betweenness centrality per module
Graph	Modularity	Maximise the modularity score [27]

[a] magic number ±7

models such as BERT [10], with bidirectional attention-based mechanism, i.e., transformers [40], can extract essential features from textual sequences and learn high-quality contextualized representations. Pre-trained BERT can be effectively employed for knowledge transfer and has produced impressive results in various downstream tasks such as open-domain question answering [45] and aspect-based sentiment analysis [43]. Therefore, pre-trained BERT embeddings can represent the conceptual model elements' terms. Therefore, we use the vector-based representations of each node to capture the natural language semantics of each node label. To evaluate the semantic similarity between two nodes, we use the cosine similarity measure between the vector embeddings of the node labels.

Finally, it is important to note that we can execute the GA using multiple objectives. However, we can also combine the objectives as a weighted sum of the objectives to create a single objective. We multiply the maximization objectives by minus one, reducing the single objective to a minimization objective. By default, we treat the weights of all objectives as equal to one.

4.3 Selection

Once we have transformed the CKG into LLE and we have an initial population of chromosomes, we need to perform a selection of candidates for the offspring generation. We use two selection operators. The tournament selection is used for choosing the chromosomes for the offspring generation when both the single- or multi-objective functions are applied. These selected chromosomes go through the altering process i.e., crossover and mutation to create new altered individuals. The roulette wheel selector is used for choosing the chromosomes for the survivor population. The altered offspring and survivor population get merged. Some individuals are removed during this merge to simulate the killing process [2].

4.4 Alteration

After selecting the chromosomes for offspring generations, we apply crossover with the selected chromosomes. The parents are randomly chosen from the offspring population to create new offsprings. The group crossover guarantees the creation of valid LLE instances. The central idea of the group crossover is to treat the ending node of the parent individuals as the central element of a module during the crossover. These central elements are then shared in the offspring instances [20]. The crossover operation generates two children, which share the ending nodes of the parents. They are inserted into the offspring population. The modules are built in the offspring from these new ending nodes after crossover. The group crossover approach also ensures that each module is a connected element, which follows the self-contained principle described in [25].

Mutation randomly takes and alters any individual from the offspring population and creates an offspring. The alteration focuses on changes in the modularization solution instance where modules will be merged, split, or elements in a module get assigned to a different module. The graft mutation operation randomly determines which of the three possible mutation types is applied. The first type of mutation divides a random module consisting of multiple elements into multiple connected submodules. The main idea involves multiple random walks to split modules into multiple connected submodules. This approach only operates inside the respective module—other modules' elements are unaffected. The next type of mutation is the combination of two random modules. In this case, only the modules with neighboring modules are used. The last mutation type moves one element in a module to a neighboring module. Elements are candidates if they are directly incident to a different module. If the element is a vertex, an incident edge must be in a different module. When the element is an edge, then one of the incident vertices must be assigned to a different module. This element is removed from the source to the target module's linked list.

4.5 Constraints and Termination

The mutation operation can produce invalid chromosomes due to randomly assigning a gene to a module. Therefore, constraints are enforced to remove invalid chromosomes. We enforce firstly that modules must be a connected subgraph to be self-contained [25], i.e., each element in the module must be reachable. Secondly, both the endpoints of an edge must be present in one of the modules. Figure 1 shows the genetic algorithm process. The process is terminated if the population's fitness reaches a pre-defined threshold based on the selected objectives. The convergence of the algorithm and the number of solutions in the Pareto set is dependent on the CKG complexity and the fitness functions. Currently, GGMF ranks all the different solutions in the Pareto set equally.

4.6 Modularization Tool

We now present the Web-based tool we developed for GGMF. Figure 4 shows the configuration parameters including the edge weights. The edge weights denote the importance of a conceptual model's different edge types of. Figure 4 shows the edge weights for a UML model, however, the edge weights as well as labels get updated depending on the type of modeling language of the input model. The user can freely customize the weights.

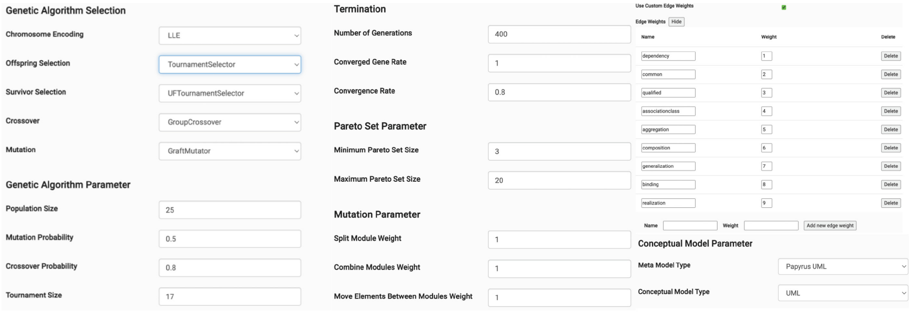


Fig. 4. Configuration parameters

The web UI is developed with Angular [1]. The web application utilizes a web component to create the user interface's visual part. In the background, a service creates an HTTP request and waits for the HTTP response. The UI is built so that it can be extended via a configuration. The default parameters, the edge weights, and the objectives are specified in a configuration file. Especially for the weights and objectives, a new entry can be added to the configuration and is immediately displayed. The rationale behind the UI was to ease the configuration of modularization experiments. The implementation of the tool can be publicly available in the GitHub Repository[2].

5 Evaluation

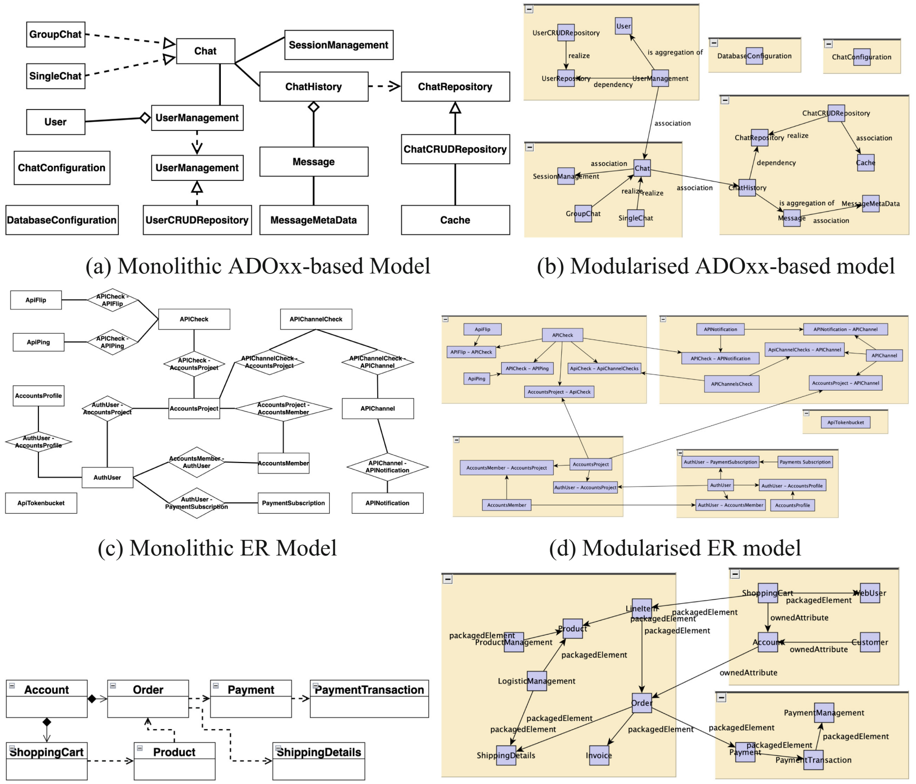
In the following, we respond to the three research questions as defined in Sect. 1 by first showing the genericity of our framework in terms of modeling languages supported (RQ1). We perform an impact analysis of the different kinds of objectives used for modularization (RQ2), and, finally, we show the performance of GGMF by comparing it to three other approaches proposed in the literature (RQ3).

5.1 Generic Modularization

Figure 5 shows the modularization result of applying our approach to different modeling languages. The figures show modularized results of the models from three different modeling languages, i.e., ER, UML and ADOxx-based UML where

[2] https://github.com/me-big-tuwien-ac-at/GGMF

UML was implemented using ADOxx development toolkit and then we used this UML implementation to create UML models in ADOxx modeling tool. The modularized results produce valid models and do not violate any constraints. These results support the feasibility of our proposed approach. Note that we conducted experiments on the dataset of 555 UML[3], 42 ER[4] models but due to limited space, we show the three representative cases in Fig. 5.



(a) Monolithic ADOxx-based Model (b) Modularised ADOxx-based model

(c) Monolithic ER Model (d) Modularised ER model

(e) Monolithic UML Model (f) Modularised UML model

Fig. 5. Three exemplary modularization experiments with different models

5.2 Objective Impact Analysis

To respond to RQ-2, i.e., how the framework caters to different requirements, we show the impact of different objectives on modularization. We show, that changing objectives affect the modularization quality, therefore, our framework

[3] https://zenodo.org/record/2585456#.YM5ziSbtb0o.
[4] https://drawsql.app/templates.

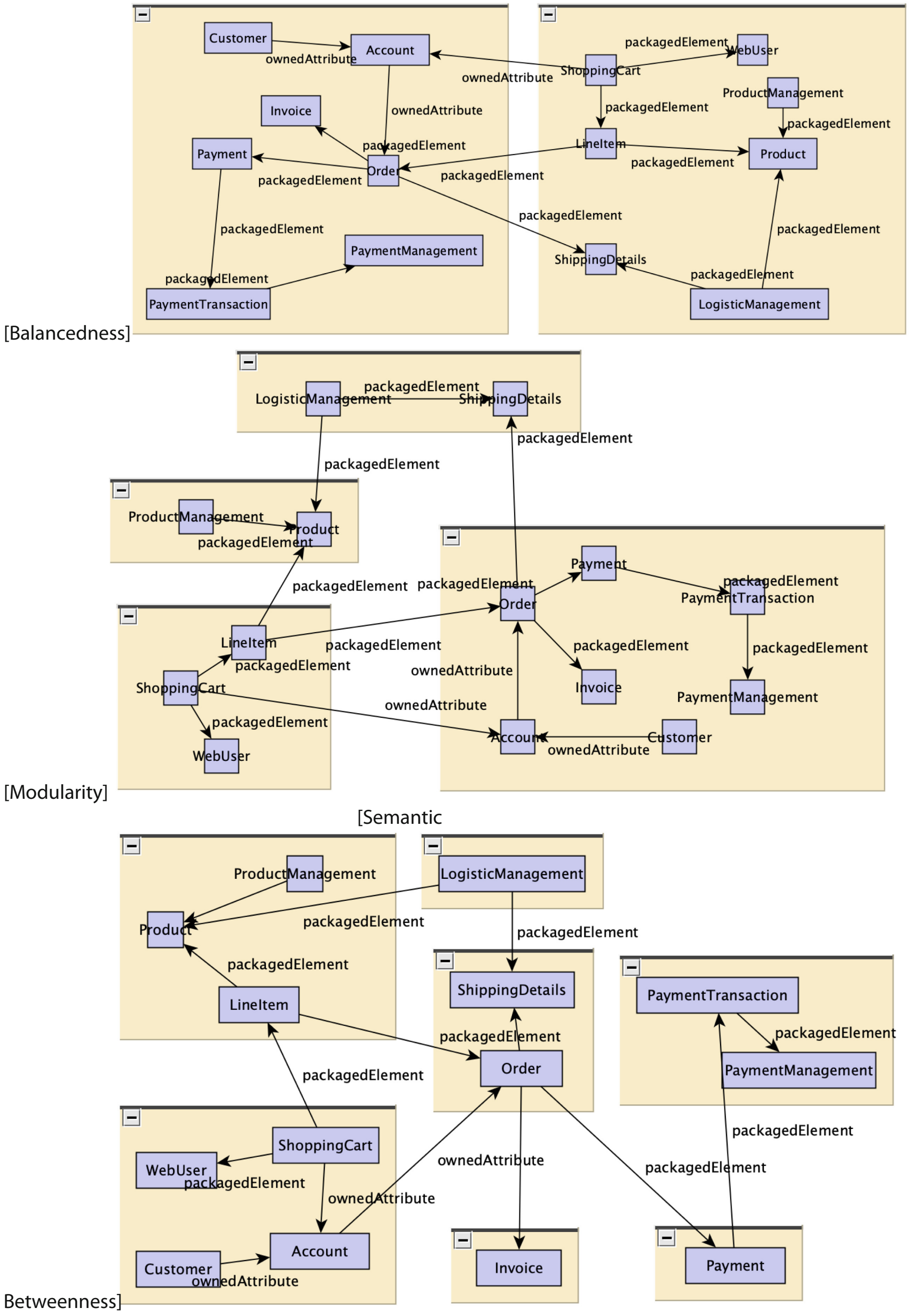


Fig. 6. Effect of different objectives on the modularization

Table 2. Modularization approaches comparison

NumNodes	Modularity				**MQ**			
	WLC	GMA	Louvain	Ours	WLC	GMA	Louvain	Ours
48	0.55	0.55	0.66	0.62	3.55	4.34	5.13	4.34
56	0.15	0.15	0.21	0.62	2	2	3.80	2
97	0.55	0.07	0.42	0.39	3.91	1.76	5.13	3.27
98	0.42	0.42	0.51	0.62	2	2	3.49	2.84
103	0.26	0.55	0.59	0.62	2	2	3.76	4.52

can support modularization in the context of changing requirements given that the user can choose appropriate objectives. Figure 6 shows the impact of using *balancedness*, *modularity*, and *semantic cohesion* as the objectives for modularization. We choose these three objectives to show the effect of different types of objectives, with balancedness focusing on individual module size, modularity focusing on the entire model, and semantic cohesion focusing on the natural language semantics of the model. We see, that using balancedness as the objective produces two modules with seven elements as per the expectation of the objective to produce modules with a size close to Miller's magic number seven [31]. In the case of modularity, we see more modules with uneven sizes. This modularization results from maximizing the modularity score as the objective given by Eq. 1. Finally, we see even more modules for semantic coupling to minimize the similarity between terms in different modules. We see that the terms in each module are semantically closer and dissimilar in different modules. Each objective has strengths and weaknesses, depending on the use case. Therefore, GGMF allows users to apply any objectives they want.

5.3 Comparative Analysis

In order to evaluate RQ-3, we compare the quality of the GGMF modularization results with three different approaches i.e., *Weighted Linking Clustering* (WLC) [24], *Graph-based Modularization* (GMA) [28], and the *Louvain* algorithm [8] on a set of five UML models. We use a distance matrix created from the adjacency matrix of the CKG as an input to WLC and a similarity matrix constructed from the distance matrix as $1 - distance$ as input to GMA. We compare the quality of the results using modularity and MQ score. Note that the purpose of this comparison is primarily to show that the results from our approach are reasonably good in the context of modularization, which can be further optimized depending on the customizations provided by GGMF based on the user requirements.

We show the comparison based on modularity and MQ Score in Table 2. We see the modularized results from our approach provide better modularity and MQ scores for the five models compared to GMA and WLC and also perform comparable to the Louvain algorithm. However, Louvain can suffer from drawbacks such as it may yield arbitrarily badly connected communities and communities may even be internally disconnected [39], which our approach explicitly avoids as part of the constraints (see Sect. 4). Moreover, we cannot apply natural language semantics-based objectives with Louvain. The results show that our GA-based optimization approach successfully finds good-quality modules of a model based on heterogeneous objectives.

6 Threats to Validity

We now elaborate on the threats to validity according to the widely accepted categories introduced by Wohlin et al. [44]. **Conclusion validity** concerns the relationship between the treatment and the outcome. We mitigated this threat by testing our framework on models of multiple modeling languages and with multiple combinations of objective functions to perform impact analysis of using a specific objective on modularization. **Internal Validity** - Parameter tuning of search algorithms is still considered an open research challenge [7]. In our work we set the configuration parameters for modularization based on experience with the modularization tool. However, we make our results reproducible with a set of configuration parameters and we expose all the configuration parameters through our web UI. **External Validity** - The quality of conceptual models used in our experiments also threatens the validity of our work. However, we mitigated this by using the dataset of models used by several works in the literature [22,23].

7 Conclusion

In this paper, we presented a generic, customizable framework for conceptual model modularization using genetic algorithm optimization techniques. We showed, that using Conceptual Knowledge Graphs as the intermediary representation of conceptual models can be used as a generic intermediary representation before applying GA-based modularization. We presented our end-to-end approach that takes as input a conceptual model and provides the modularized model. We showed the feasibility of generic modularization by executing our approach on models from three modeling languages i.e., ER, UML and ADOxx-based models. Different requirements require targeting different objectives, therefore, we showed the effect of using different objectives on the modularization results. GGMF allows extending the list of objectives that can be used by the GA for optimization, thereby supporting modularization depending on different requirements. We compared the quality of the solutions produced by our approach with two other approaches. The results show that our approach provides modules of comparable quality. Finally, we showed the interface of the web-based modularization tool that we developed, which allows users to configure the parameters for the modularization and also adjust the importance (weights) of different types of edges present in a model, which allows using a weighted contribution of an edge while evaluating the quality of a module during optimization. Consequently, GGMF enables an entirely new level of flexibility and customizability of GA-based model modularization which is also applicable for non-technical users.

In the future, we focus on the following potential improvements of GGMF: *i*) advanced qualitative evaluation of the resulting modules in relation to the initial modularization goals, which is currently quantitatively evaluated using the fitness scores of the involved fitness functions; *ii*) statistical-analysis based GGMF configuration parameter values recommendation for diverse models; and *iii*) comparative analysis with ML-based modularization techniques that use structure and semantics for modularization.

References

1. Angular. https://angular.io/. Accessed 30 July 2022
2. Jenetics. https://jenetics.io/. Accessed 09 July 2022
3. Ali, S.J., Guizzardi, G., Bork, D.: Enabling representation learning in ontology-driven conceptual modeling using graph neural networks. In: Indulska, M., Reinhartz-Berger, I., Cetina, C., Pastor, O. (eds.) CAiSE 2023. LNCS, vol. 13901, pp. 278–294. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34560-9_17
4. Andritsos, P., Tzerpos, V.: Information-theoretic software clustering. IEEE Trans. Software Eng. **31**(2), 150–165 (2005)
5. Bae, J.H., Lee, K., Chae, H.S.: Modularization of the UML metamodel using model slicing. In: Fifth International Conference on Information Technology: New Generations (ITNG 2008), pp. 1253–1254 (2008). https://doi.org/10.1109/ITNG.2008.179
6. Bavota, G., Carnevale, F., De Lucia, A., Di Penta, M., Oliveto, R.: Putting the developer in-the-loop: an interactive GA for software re-modularization. In: Fraser, G., Teixeira de Souza, J. (eds.) SSBSE 2012. LNCS, vol. 7515, pp. 75–89. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33119-0_7
7. Bill, R., Fleck, M., Troya, J., Mayerhofer, T., Wimmer, M.: A local and global tour on MOMoT. Softw. Syst. Model. **18**, 1017–1046 (2019)
8. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
9. Bork, D., Garmendia, A., Wimmer, M.: Towards a multi-objective modularization approach for entity-relationship models. In: ER Forum, Demo and Poster 2020, pp. 45–58. CEUR (2020)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Doran, P., Tamma, V., Iannone, L.: Ontology module extraction for ontology reuse: an ontology engineering perspective. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 61–70 (2007)
12. Figueiredo, G., Duchardt, A., Hedblom, M.M., Guizzardi, G.: Breaking into pieces: an ontological approach to conceptual model complexity management. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS), pp. 1–10. IEEE (2018)

13. Freeman, L.: A set of measures of centrality based on betweenness. Sociometry **40**, 35–41 (1977). https://doi.org/10.2307/3033543
14. Glaser, P.L., Sallinger, E., Bork, D.: Model-based construction of enterprise architecture knowledge graphs (2022, under review)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, New York (1989)
16. Hinkel, G., Strittmatter, M.: On using sarkar metrics to evaluate the modularity of metamodels. In: Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development, pp. 253–260. Springer, Cham (2017)
17. Hinkel, G., Strittmatter, M.: Predicting the perceived modularity of MOF-based metamodels. In: 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018), Funchal, P, 22–24 January 2018, pp. 48–58. SciTePress (2018)
18. Kang, D., Xu, B., Lu, J., Chu, W.: A complexity measure for ontology based on UML. In: Proceedings of 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, FTDCS 2004, pp. 222–228 (2004)
19. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: past, present, and future. Multimedia Tools Appl. **80**(5), 8091–8126 (2020). https://doi.org/10.1007/s11042-020-10139-6
20. Korkmaz, E.: Multi-objective genetic algorithms for grouping problems. Appl. Intell. **33**, 179–192 (2010). https://doi.org/10.1007/s10489-008-0158-3
21. LeClair, A., Marinache, A., El Ghalayini, H., MacCaull, W., Khedri, R.: A review on ontology modularization techniques-a multi-dimensional perspective. IEEE Trans. Knowl. Data Eng. **35**(5), 4376–4394 (2022)
22. López, J.A.H., Cánovas Izquierdo, J.L., Cuadrado, J.S.: Modelset: a dataset for machine learning in model-driven engineering. Softw. Syst. Model. 1–20 (2022)
23. López, J.A.H., Cuadrado, J.S.: An efficient and scalable search engine for models. Softw. Syst. Model. **21**(5), 1715–1737 (2022)
24. Maqbool, O., Babri, H.: Hierarchical clustering for software architecture recovery. IEEE Trans. Software Eng. **33**(11), 759–780 (2007)
25. Moody, D.L., Flitman, A.: A methodology for clustering entity relationship models — a human information processing approach. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) ER 1999. LNCS, vol. 1728, pp. 114–130. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-47866-3_8
26. Mu, L., Sugumaran, V., Wang, F.: A hybrid genetic algorithm for software architecture re-modularization. Inf. Syst. Front. **22**, 1133–1161 (2020)
27. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006). https://doi.org/10.1073/pnas.0601602103
28. Pourasghar, B., Izadkhah, H., Isazadeh, A., Lotfi, S.: A graph-based clustering algorithm for software systems modularization. Inf. Softw. Technol. **133**, 106469 (2021)
29. Prajapati, A., Kumar Chhabra, J.: Optimizing software modularity with minimum possible variations. J. Intell. Syst. **29**(1), 1135–1150 (2020). https://doi.org/10.1515/jisys-2018-0231
30. Proper, H.A., Guizzardi, G.: Modeling for enterprises; let's go to RoME ViA RiME. Hand **1**, 3 (2022)
31. Saaty, T.L., Ozdemir, M.S.: Why the magic number seven plus or minus two. Math. Comput. Model. **38**(3–4), 233–244 (2003)
32. Sarkar, S., Kak, A.C., Maskeri Rama, G.: Metrics for measuring the quality of modularization of large-scale object-oriented software. IEEE Trans. Softw. Eng. **34**(05), 700–720 (2008). https://doi.org/10.1109/TSE.2008.43

33. Saruladha, K., Aghila, G., Sathiya, B.: Neighbour based structural proximity measures for ontology matching systems. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 1079–1085 (2012)
34. Sequeda, J., Lassila, O.: Designing and building enterprise knowledge graphs. Synth. Lect. Data Semant. Knowl. **11**(1), 1–165 (2021)
35. Singh, P., Jonkers, H., Iacob, M.E.: Modeling value creation with enterprise architecture. In: ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems, vol. 3, pp. 343–351 (2014)
36. Smajevic, M., Bork, D.: Towards graph-based analysis of enterprise architecture models. In: Ghose, A., Horkoff, J., Silva Souza, V.E., Parsons, J., Evermann, J. (eds.) ER 2021. LNCS, vol. 13011, pp. 199–209. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-89022-3_17
37. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 289–303. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30475-3_21
38. Tabrizi, A.H.F., Izadkhah, H.: Software modularization by combining genetic and hierarchical algorithms. In: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), pp. 454–459. IEEE (2019)
39. Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Sci. Rep. **9**(1), 5233 (2019)
40. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
41. Villegas Niño, A.: A filtering engine for large conceptual schemas. Doctoral thesis (2013)
42. Vragović, I., Louis, E.: Network community structure and loop coefficient method. Phys. Rev. E **74**, 016105 (2006). https://doi.org/10.1103/PhysRevE.74.016105
43. Wang, Y., Chen, Q., Wang, W.: Multi-task BERT for aspect-based sentiment analysis. In: 2021 IEEE International Conference on Smart Computing, pp. 383–385. IEEE (2021)
44. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29044-2
45. Yang, W., et al.: End-to-end open-domain question answering with bertserini. arXiv preprint arXiv:1902.01718 (2019)

MARTSIA: Enabling Data Confidentiality for Blockchain-Based Process Execution

Edoardo Marangone[1](✉), Claudio Di Ciccio[1], Daniele Friolo[1], Eugenio Nerio Nemmi[1], Daniele Venturi[1], and Ingo Weber[2,3]

[1] Sapienza University of Rome, Rome, Italy
{marangone,diciccio,friolo,nemmi,venturi}@di.uniroma1.it
[2] Technical University of Munich, School of CIT, Munich, Germany
ingo.weber@tum.de
[3] Fraunhofer Gesellschaft, Munich, Germany

Abstract. Multi-party business processes rely on the collaboration of various players in a decentralized setting. Blockchain technology can facilitate the automation of these processes, even in cases where trust among participants is limited. Transactions are stored in a ledger, a replica of which is retained by every node of the blockchain network. The operations saved thereby are thus publicly accessible. While this enhances transparency, reliability, and persistence, it hinders the utilization of public blockchains for process automation as it violates typical confidentiality requirements in corporate settings. In this paper, we propose *MARTSIA*: A *Multi-Authority Approach to Transaction Systems for Interoperating Applications*. MARTSIA enables precise control over process data at the level of message parts. Based on Multi-Authority Attribute-Based Encryption (MA-ABE), MARTSIA realizes a number of desirable properties, including confidentiality, transparency, and auditability. We implemented our approach in proof-of-concept prototypes, with which we conduct a case study in the area of supply chain management. Also, we show the integration of MARTSIA with a state-of-the-art blockchain-based process execution engine to secure the data flow.

Keywords: Multi-Authority Attribute Based Encryption · Distributed Ledger Technology · InterPlanetary File System

1 Introduction

Enterprise applications of blockchain technology are gaining popularity because it enables the design and implementation of business processes involving many parties with little mutual trust, among other benefits [36,40]. Standard blockchains yield the capability of enabling cooperation between potentially untrusting actors through transparency: relevant data is made available to all participants of a blockchain network, and hence can be verified by anyone, thereby removing the need for trust [42]. In combination with the high-integrity permanence of data and non-repudiability of transactions offered by the technology, blockchains can be used to realize trustworthy protocols.

However, in multi-party business settings with scarce mutual trust, the involved parties typically have a strong need to keep certain data hidden from some of the business

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 58–76, 2024.
https://doi.org/10.1007/978-3-031-46587-1_4

partners, and even more so from most other participants in a blockchain network. In fact, fulfilling security and privacy requirements is a key obstacle when it comes to the adoption and implementation of blockchain technology in general [14,43]. Corradini et al. [10] confirm the importance of security and privacy considerations for the specific case of process execution on blockchain.

Simple cryptographic solutions face severe downsides, as discussed in the following. First, the authors of [10] note that simply encrypting the contents of messages (payload), as previously proposed in the literature, does not guarantee the confidentiality of the information. Using synchronous encryption requires sharing a decryption key among process participants, and thus does not allow the sender of the data to selectively control access to different parts of a single message. Using asynchronous encryption and encrypting a message with the public key of the recipient requires the sender to create multiple copies of each message (one for each intended reader), which means that the sender can send *different* pieces of information to each participant – i.e., integrity is lost. Other proposed solutions address the issue via perimeter security: read access to the (relevant parts of) a blockchain is limited, e.g., by using channels on Hyperledger Fabric or similar [42, Ch. 2], or using private blockchains. However, this approach suffers from the same downsides as the use of synchronous encryption above. Also, permissioned platforms require the presence of trusted actors with the privileged role of managing information exchange *and* the right to be part of the network. In summary, most of the previous approaches offer "all-or-nothing" access: either all participants in some set can access the information in a message, or they receive only private messages and integrity of the data sent to multiple recipients is lost. In previous work [24], we introduced an early approach to control data access at a fine-granular level. However, the architecture relied on a central node for forging and managing access keys, thus leading to easily foreseeable security issues in case this single component were to be compromised or byzantine. Also, its integration with process management systems was yet to be verified. With the objective of overcoming these limitations, we have revised the entire approach from its foundations and devised the new solution we present here.

In this paper, we propose a Multi-Authority Approach to Transaction Systems for Interoperating Applications (MARTSIA). In MARTSIA, encrypted data is persisted in decentralized storage, which is connected to a public permissionless blockchain system supporting process execution. Data owners define access policies to regulate which users are able to view specific parts of the information. No central authority can generate decryption keys alone. The encryption and decryption of messages are left to the individual nodes. To attain the desired characteristics, this approach employs hybrid encryption in combination with the Ciphertext-Policy variant of Multi-Authority Attribute-Based Encryption (henceforth, MAABE for the sake of conciseness), smart contracts, and InterPlanetary File System (IPFS). In our evaluation, we show the integration of our implemented prototype with Caterpillar [22], a state-of-the-art process execution engine, to demonstrate how our approach can complement a business process management system to secure its data flow.

In the following, Sect. 2 presents a running example, to which we will refer throughout the paper, and illustrates the problem we tackle. Section 3 outlines the fundamental notions that our solution is based upon. In Sect. 4, we describe our approach in detail. In

Sect. 5, we present our proof-of-concept implementation and the results of the experiments we conducted therewith. Section 6 reviews related work before Sect. 7 concludes the paper and outlines future works.

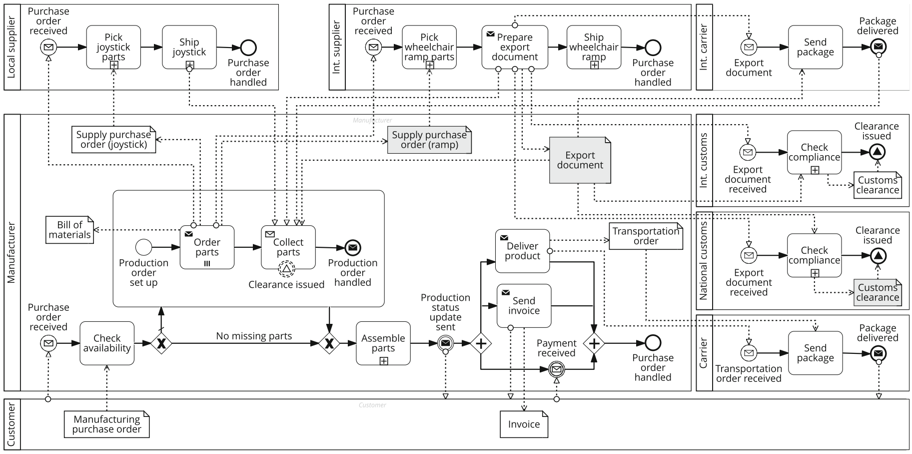


Fig. 1. A multi-party process for the assembly of special car parts.

2 Example, Problem Illustration, and Requirements

Figure 1 shows a Business Process Model and Notation (BPMN) collaboration diagram illustrating a supply chain in the automotive area: the production of a special car for a person with paraplegia. The two base components for that car are a joystick (to turn, accelerate, brake) and a wheelchair ramp to let the person get into the vehicle.

A new process instance is initiated when a *customer* places an order for a modified car from a *manufacturer*. The *manufacturer* then checks the availability of joystick parts and wheelchair ramps in the warehouse, to order the missing ones from a local *joystick supplier* and an international *wheelchair ramp supplier*, respectively. Once all ordered parts have been collected, the suppliers prepare the packages with the products for delivery. The *international customs* verifies the *document* of the international supplier and issues custom clearance once the compliance verification is completed successfully. The *carrier* of the international supplier checks the documents and delivers the package to the *manufacturer* with an international shipment procedure. Upon receipt of the parts, the *manufacturer* proceeds with the assembly process. After informing the *customer* about the progress of the production process, the *manufacturer* sends an invoice and requests a carrier to deliver the package. The process is completed with the delivery of the ordered product.

Throughout the paper, we will refer to this scenario as a running example. In particular, we will focus on the information artifacts in Table 1 (marked with a gray background color in Fig. 1), namely: (1) the purchase order of the manufacturer, (2) the export document of the international supplier, and (3) the national customs clearance.

Table 1. An excerpt of the information artifacts exchanged in Fig. 1

Message	Sender	Data	Recipients

Supply purchase order (ramp)	Manufacturer	`Company_name: Alpha`
		`Address: 38, Alpha street`
		`E-mail: cpny.alpha@mail.com`
		`Price: $5000`
			International supplier

Export document	International supplier	`Manufacturer_company: Beta`
		`Delivery_address: 82, Beta street`
		`E-mail: mnfctr.beta@mail.com`
		`Ramp run: 3`
		`Kickplate: 12`
		`Handrail: 7`
		`Baluster: 30`
		`Guardrail: 25`
		`Amount_paid: $5000`
			Manufacturer
			National customs
			International customs
			International carrier

		`Fundamental_workers_rights: Ok`
		`Human_rights: Ok`
		`Protection_biodiversity_and_ecosystems: Ok`
		`Protection_water_and_air: Ok`
		`Combatting_climate_change: Ok`
			National customs
			International customs

		`Manufacturer_company: Beta`
		`Address: 78, Beta street`
		`Order_reference: 26487`
			Manufacturer

		`Invoice_ID: 101711`
		`Billing_address: 34, Gamma street`
		`Gross_total: $5000`
		`Company_VAT: U12345678`
		`Issue_date: 2022-05-12`
			Manufacturer
			National customs
			International customs

Customs clearance	National customs	`Tax_payment: confirmed`
		`Conformity_check: passed`
		`Date: 2022-05-10`
		`Sender: Beta`
		`Receiver: Alpha`
			Manufacturer
			International customs

Table 2. Requirements and corresponding actions in the approach

	Requirement	CAKE [24]	MARTSIA	See
R1	Access to parts of messages should be controllable in a fine-grained way (attribute level), while integrity is ensured	✓	✓	Sects. 4.2 and 5
R2	Information artifacts should be written in a permanent, tamper-proof and non-repudiable way	✓	✓	Sects. 4.1 and 5
R3	The system should be independently auditable with low overhead	✓	✓	Sects. 4.1 and 5
R4	The decryption key should only be known to the user who requested it	×	✓	Sects. 4.1 and 5
R5	The decryption key should not be generated by a single trusted entity	×	✓	Sects. 4.1 and 5
R6	The approach should integrate with control-flow management systems	×	✓	Sects. 4.2 and 5

The export document encloses multiple records, namely (2.a) the international shipment order, (2.b) the Corporate Sustainability Due Diligence Directive (CSDD), (2.c) the reference to the order, and (2.d) the invoice. These records are meant to be accessed by different players. The shipment order should only be accessible by the international carrier and the two customs bodies, the CSDD can only by read by the customs authorities, the order reference is for the manufacturer, and the invoice is for the manufacturer and the customs bodies. Differently from the purchase order and the customs clearance (messages 1 and 3), the four above entries (2.a to 2.d) are joined in a single document for security reasons: separate messages could be intercepted and altered, replaced or forged individually. Once they are all part of a single entity, every involved actor can validate all the pieces of information. Ideally, in a distributed fashion every node in the network could be summoned to attest to the integrity of that document. However, the need for separation of indicated recipients demands that only a selected group of readers be able to interpret the parts that are specifically meant for them (see the rightmost column of Table 1). In other words, though *visible* for validation, the data artifact should not be interpretable by everyone. The other actors should attest to data encrypted as in Table 3. This aspect gives rise to one of the requirements we discuss next.

Requirements. In recent years, there has been a surge in research on blockchain-based control-flow automation and decision support for processes (see [36] for an overview). Typically, information shared by actors in a collaborative process is commercial-in-confidence, i.e., shared only with the parties that need access to it, and who are in turn expected to not pass the information on. Our research complements this work by focusing on secure information exchange among multiple parties in a collaborative though partially untrusted scenario.

Table 2 lists the requirements stemming from the motivating use case that drives our approach and a research project in which two authors of this paper are involved.[1] The table highlights the limitations of our past work [24] that we overcome and indicates the sections in which we discuss the action taken to meet them. Different parties should be granted access to different sections of a confidential information source (**R1**, as in the case of the export document in our motivating scenario). The information source should remain available, immutable, and accountability should be granted for subsequent validations and verifications (**R2**, as for the check of the invoice by customs and, more in general, for process mining and auditing [17]), without major overheads (**R3**) for practical feasibility. In a distributed scenario such as that of the process in Sect. 4, where multiple authorities and actors are involved, it is necessary to secure the infrastructure by avoiding that any party can acquire (**R4**) or forge (**R5**) decryption keys alone. Finally, our approach should complement existing process execution engines to intercept and secure the data flow that characterized multi-party collaborations (**R6**). Next, we discuss the background knowledge that our approach is based on.

3 Background

Distributed Ledger Technologies (DLTs), and specifically programmable blockchain platforms, serve as the foundation for our work together with Multi-Authority Attribute-Based Encryption (MA-ABE). Here, we explain the basic principles underneath these building blocks.

Distributed Ledger Technologies. DLTs are protocols that allow for the storage, processing, and validation of transactions among a network of peers without the need for a central authority or intermediary. These transactions are timestamped and signed cryptographically, relying on asymmetric or public key cryptography with a pair of a private and a public key. In DLTs, every user has an account with a unique address, associated with such a key pair. The shared transaction list forms a ledger that is accessible to all participants in the network. A **blockchain** is a specific type of DLT in which transactions are strictly ordered, grouped into blocks, and linked together to form a chain. DLTs, including blockchains, are resistant to tampering due to the use of cryptographic techniques such as hashing (for the backward linkage of blocks to the previous one), and the distributed validation of transactions. These measures ensure the integrity and

[1] Cyber 4.0 project BRIE: https://brie.moveax.it/en. Accessed: 09 June 2023.

security of the ledger. Blockchain platforms come endowed with consensus algorithms that allow the distributed networks to reach eventual consistency on the content of the ledger [27]. Public blockchains like Ethereum [41] charge fees for the inclusion and processing of transactions. Ethereum supports expressive smart contracts, which are user-defined programs. They are deployed and invoked through transactions, i.e., their code is stored on chain and executed by many nodes in the network. Outcomes of contract invocations are part of the blockchain consensus, thus verified by the blockchain system and fully traceable. The execution of smart contract code, like transactions, incurs costs measured as *gas* in the Ethereum platform. Gas cost is based on the complexity of the computation and the amount of data exchanged and stored. To lower the costs of invoking smart contracts, external Peer-to-peer (P2P) systems are often utilized to store large amounts of data [42]. One of the enabling technologies is **InterPlanetary File System (IPFS)**,[2] a distributed system for storing and accessing files that utilizes a Distributed Hash Table (DHT) to scatter the stored files across multiple nodes. Like DLTs, there is no central authority or trusted organization that retains control of all data. IPFS uses content-addressing to uniquely identify each file on the network. Data stored on IPFS is linked to a resource locator through a hash, which–in a typical blockchain integration–is then sent to a smart contract to be stored permanently on the blockchain [21]. In a multi-party collaboration setting like the one presented in Sect. 2, the blockchain provides an auditable notarization infrastructure that certifies transactions among the participants (e.g., purchase orders or customs clearances). Smart contracts ensure that the workflow is carried out as agreed upon, as described in [12,25,40]. Documents like purchase orders, transportation orders, and customs clearances can be stored on IPFS and linked to transactions that report on their submission. However, data is accessible to all peers on the blockchain. To take advantage of the security and traceability of the blockchain while also controlling access to the stored information, it is necessary to encrypt the data and manage read and write permissions for specific users.

Attribute-Based Encryption (ABE). ABE is a form of public key encryption in which the *ciphertext* (i.e., an encrypted version of a *plaintext* message) and the corresponding decryption key are connected through attributes [6,35]. In particular, Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [4,20] associates each potential user with a set of attributes. Policies are expressed over these attributes using propositional literals that are evaluated based on whether a user possesses a particular property. In the following, we shall use the teletype font to format attributes and policies. For example, user `0xB0...1AA1` is associated with the attributes `Supplier`, to denote their role, and `43175279`, to specify the process instance number they are involved in (the *case id*). For the sake of brevity, we omit from the attribute name that the former is a role and the latter a process instance identifier (e.g., `Supplier` in place of `RoleIsSupplier` or `43175279` instead of `InvolvedInCase43175279`) as we assume it is understandable from the context. Policies are associated with ciphertexts and expressed as propositional formulae on the attributes (the literals) to determine whether a user is granted access (e.g., `Carrier or Manufacturer`).

As argued in the introduction, one goal of this work is to move away from a single source of trust (or failure); thus, we consider multi-authority methods. To decrypt and

[2] ipfs.tech. Accessed: 09 June 2023.

access the information in a ciphertext, a user requires a dedicated key. With Multi-Authority Attribute-Based Encryption (MA-ABE), every authority creates a part of that key, henceforth *decryption key (dk)*. A *dk* is a string generated via MA-ABE on the basis of *(i)* the user attributes, and *(ii)* a secret key of the authority. To generate the secret key (coupled with a public key), the authority requires public parameters composed of a sequence of pairing elements that are derived from a pairing group via Elliptic Curve Cryptography (ECC). Due to space restrictions, we cannot delve deeper into the notions of pairing groups and pairing elements. We refer to [6,26] for further details. Once the user has obtained a *dk* from every required authority, it merges them obtaining the *final decryption key (fdk)* to decrypt the message.

In the Cypertext-Policy variant of MA-ABE, a ciphertext for a given message is generated from the public parameters, the public keys of all the authorities, and a policy. In our context, users are process participants, messages are the data artefacts exchanged during process execution, ciphertexts are encrypted versions of these artefacts, policies determine which artefacts can be accessed by which users, and keys are the tools granted to process parties to try to access the artefacts. In the following sections, we describe how we combine the use of blockchain and the Cypertext-Policy variant of MA-ABE to create an access control architecture for data exchanges on the blockchain that meets the requirements listed in Table 2.

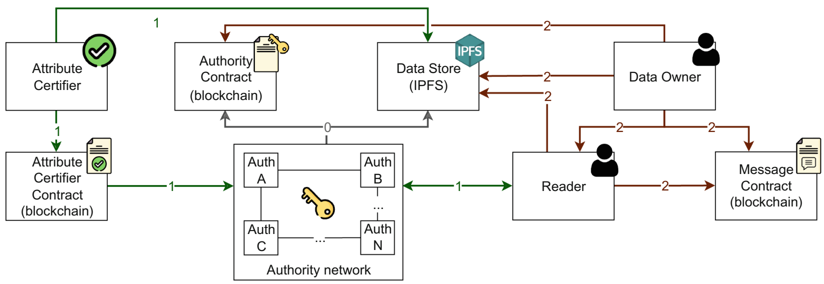


Fig. 2. The key components and their interactions in the MARTSIA approach

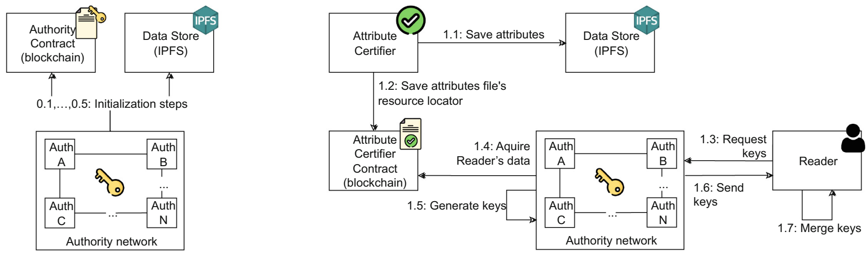
4 The MARTSIA Approach

In this section, we describe our approach, named Multi-Authority Approach to Transaction System for Interoperating Applications (MARTSIA). We begin by examining the collaboration among its core software components, and then illustrate the data structures they handle.

4.1 Workflow

Figure 2 illustrates the main components of our architecture and their interactions. The involved parties are: **the Attribute Certifier** specifying the attributes characterizing the potential readers of the information artifacts; we assume the Attribute Certifiers to hold a blockchain account; different Attribute Certifiers may attest to different pieces of information about potential readers; **the Data Owner** encrypting the information artifacts (henceforth also collectively referred to as *plaintext*) with a specific access policy (e.g., the manufacturer who wants to restrict access to the purchase orders to the sole intended parties, i.e., the suppliers); we assume the Data Owner to hold a blockchain account and a Rivest-Shamir-Adleman (RSA) [33] secret/public-key pair; **Readers** interested in some of the information artifacts (e.g., the manufacturer, the joystick supplier, and the wheelchair ramp supplier); we assume the Readers to hold a blockchain account, a RSA secret-key/public-key pair, and a global identifier (GID) that uniquely identifies them; **the Authorities** that calculate their part of the secret key for the Reader; **the Data Store,** a P2P repository based on IPFS. IPFS saves all exchanged pieces of information in a permanent, tamper-proof manner creating a unique content-based hash as resource locator for each of them; **the Smart Contracts** used to safely store and make available the resource locators to the ciphertext saved on the Data Store (**Message Contract**), the information about potential readers (**Attribute Certifier Contract**), and the data needed by the authorities to generate the public parameters (**Authority Contract**).

We divide our approach in three main phases, which we discuss in detail next: initialization (Fig. 3(a)), key management (Fig. 3(b)), and data exchange (Fig. 4). In the following, the numbering scheme corresponds to the labels in Figs. 2 to 4.



(a) Authority initialization (b) Key management

Fig. 3. Authority initialization and key management phases in MARTSIA

0: Initialization. Here we focus on the network of authorities, as depicted in Fig. 3(a). The initialization phase consists of the following five steps. (**0.1**) First, each authority creates a separate file with the metadata of all the authorities involved in the process. [3] Authorities are responsible for the setting of public parameters that are crucial to all the algorithms of MA-ABE. Therefore, we have redesigned the public parameter generation

[3] Notice that metadata are known to all the authorities and all the actors involved in the process. Therefore, non-malicious authorities are expected to create an identical file. The (same) hash is thus at the basis of the resource locator. As a consequence, anyone can verify whether the authorities behave properly in this step by checking that the resource locators are equal, with no need to load the file from the Data Store.

program as a Multi-party Computation (MPC) protocol [8,9] to guarantee full decentralization. More specifically, we adapt a commit-then-open coin-tossing protocol [5] as follows to generate a random pairing element, that is, the core piece of data described in [34] for MA-ABE implementation. (**0.2**) Each authority posts on the blockchain the hash of a locally generated random pairing element by invoking the Authority Contract. (**0.3**) After all the hashes are publicly stored, each authority posts the *opening*, namely the previously hashed pairing element in-clear, completing the commit-then-open coin-tossing protocol introduced before. (**0.4**) Then, every authority *(i)* verifies that all the hashes of the pairing elements match the respective openings, *(ii)* independently combines all posted openings via bitwise XOR, and *(iii)* uses the output of this operation (the *final shared pairing element*) to calculate the set of public parameters as illustrated in [34]. (**0.5**) Each authority generates its own public-key/secret-key pair by using the authority key generation algorithm of MA-ABE. To enable full decentralization and notarization, we resort to the Data Store to save the output of all actions (**0.1** to **0.5**) and the Authority Contract to keep track of the corresponding resource locators.

1: Key Management. The key management phase is comprised of the following steps, as illustrated in Fig. 3(b): (**1.1**) The Attribute Certifiers save the attributes and the identifying blockchain account addresses of the Readers on the Data Store and (**1.2**) the corresponding resource locator on the Attribute Certifier Contract so as to make them publicly verifiable on chain. To this end, every Attribute Certifier operates as a push-inbound oracle [2], storing on chain the attributes that determine the role of the Reader and, optionally, the list of process instances in which they are involved. For example, an Attribute Certifier stores on chain that `0x82...1332` is the address of a user that holds the `Manufacturer` role and participates in the process identified by `43175279`. Another Attribute Certifier registers that `0xB0...1AA1` and `0x9E...C885` are Readers both endowed with the `Supplier` and `43175279` attributes, though the former is `National` and the latter `International`. Readers' attributes are stored on a public blockchain for verifiability. However, notice that Readers are referred to by their public addresses, thus keeping pseudonimity. Also, the attribute names are strings that we keep intuitively understandable for the sake of readability in this paper, yet only serve as propositional symbols for the encoding of policies. Obfuscation techniques can thus be seamlessly applied though their discussion goes beyond the scope of this paper. Whenever a Reader (e.g., the international customs) wants to access the data of a message (e.g., the sections of interest in the Document), they operate as follows: (**1.3**) They request a key to all the authorities, passing the identifying GID as input (we enter the

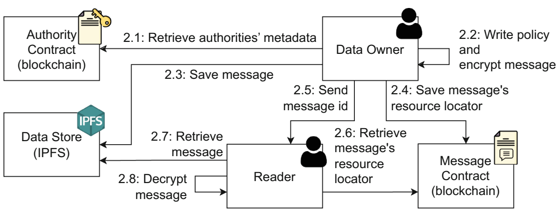


Fig. 4. The data exchange phase in MARTSIA

detail of this passage later); (**1.4**) Each authority seeks the Reader data (the blockchain address and attributes), and obtains them from the Attribute Certifier Contract; (**1.5**) Equipped with these pieces of information alongside the public parameters, the secret key, and the user's GID, each authority produces a MA-ABE decryption key (*dk*) for the Reader, and (**1.6**) sends it back. Once all *dk*s are gathered from the authorities, (**1.7**) the Reader can merge them to assemble their own *fdk*. Notice that none of the Authorities can create the *fdk* alone (unless specified as such), thus meeting **R5**; no user other than the intended Reader can obtain the key (**R4**). The key management phase can be interleaved with the Data Exchange phase (below).

A Note on the Security of Key Requests. Maliciously obtaining the *fdk* of another Reader is a high security threat. We decompose this issue in two challenges. First, we want to avoid that information exchanged between a Reader and an Authority is intercepted by other parties. To this end, we convey every communication in step **1.3** through a separate client-server Secure Sockets Layer (SSL) connection between the Reader and each of the authorities. To avoid any false self-identification as a Reader through their GID, we include a handshake preliminary phase in the protocol. It starts with every Authority (server) sending a random value (or *challenge*) to the Reader (client). The latter responds with that value signed with their own RSA private key, so as to let the invoked components verify their identity with the caller's public key.

2: Data Exchange. Figure 4 presents the operations carried out for information storage and access. As a preliminary operation, the Data Owner verifies that the hash links of the files with metadata and public parameters posted by all the authorities are equal to one another to ascertain their authenticity.(See footnote 3) Then, data is transferred from the Data Owner to the Readers through the following steps. (**2.1**) The Data Owner retrieves the authorities' public keys and public parameters from the blockchain. (**2.2**) Then, they write a policy. Notice that standard MA-ABE sets a maximum length for the input files. In a business process context, this limitation would undermine practical adoption. To cater for the encryption of arbitrary-size plaintexts, we thus resort to a two-staged hybrid encryption strategy [11]. First, the Data Owner encrypts via MA-ABE a randomly generated symmetric key (of limited size, e.g., `b'3:go+s...x2g='`) with the authorities' public keys and the policy (obtaining, e.g., `eJytm1...eaXV2u`). Afterwards, it encrypts the actual information artifact (of any size) via symmetric key encryption scheme [28] using that symmetric key. In our example scenario, then, the manufacturer does not encrypt via MA-ABE the supply purchase order, but the key through which that document is encrypted (and decryptable). (**2.3**) Thereupon, the Data Owner (e.g., the manufacturer) saves the encrypted symmetric key and information artifact (plus additional metadata we omit here for the sake of clarity and detail in Sect. 4.2) in one file on the Data Store, (**2.4**) sends the file's resource locator to the Message Contract, and (**2.5**) transmits the unique message ID (e.g., 22063028) assigned to the file to the Reader (e.g., the supplier). As the information artifact is on the Data Store and its resource locator saved on chain, it is written in a permanent, tamper-proof and non-repudiable way, thus meeting requirement **R2**. Equipped with their own *fdk*, the Reader can begin the message decryption procedure. (**2.6**) At first, the Reader retrieves the resource locator of the message from the Message Contract. (**2.7**) Then, once the Reader obtains the ciphertext from the Data Store, they pass it as input alongside the public parameters

Table 3. Example of messages stored by MARTSIA upon encryption

Message	Metadata	Header	Body (slices)
Supply purchase order (ramp)	sender: 0x82[...]1332 process_instance_id: 43175279 message_id: 22063028	EncryptedKey: eJytm1[...]eaXV2u Fields: {CV8=[...]6w==, p84W[...]aw==, avNL[...]lw==, AOo=[...]mQ==}	CV8=[...]6w==: Ruk/[...]QQ== p84W[...]aw==: 1UTk[...]mA== avNL[...]lw==: VT+D[...]JQ== AOo=[...]mQ==: ZUE/[...]6w==
Export document	sender: 0x9E[...]C885 process_instance_id: 43175279 message_id: 15469010	SliceId: 62618638 EncryptedKey: eJytWU[...]Bmn4k= Fields: {u7o=[...]iw==, TacL[...]IQ==, EhrB[...]lw==, Tiba[...]Wg==, xJ+6[...]SQ==, RPn7[...]Zz==, T3Wq[...]eK==, QXjN[...]LS==, Jmk8[...]fL==}	u7o=[...]iw==: py8l[...]mA== TacL[...]IQ==: MKNH[...]uQ== EhrB[...]lw==: 5bz0[...]bg== Tiba[...]Wg==: JahD[...]5Q== xJ+6[...]SQ==: 3aIm[...]kg== RPn7[...]Zz==: nRNh[...]GA== T3Wq[...]eK==: PxRk[...]0g== QXjN[...]LS==: Lfhk[...]Ug== Jmk8[...]fL==: Gj43[...]yg==
		SliceId: 19756540 EncryptedKey: eJytm0[...]wtYFo= Fields: {ZOSH[...]9B==, BZ8l[...]KY==, QeUy[...]WT==, rtdZ[...]hf==, 8kuG[...]Ts==}	ZOSH[...]9B==: sctC[...]nQ== BZ8l[...]KY==: 5Gln[...]Pg== QeUy[...]WT==: k49+[...]kA== rtdZ[...]hf==: lRLd[...]tm== 8kuG[...]Ts==: 25E0[...]uc==
		SliceId: 12191034 EncryptedKey: eJytWU[...]sGlU4= Fields: {t8gr[...]QQ==, yuwd[...]vg==, 1K1d[...]zQ==}	t8gr[...]QQ==: ZJ1v[...]5A== yuwd[...]vg==: 7XpN[...]4A== 1K1d[...]zQ==: +QbM[...]Cw==
		SliceId: 98546521 EncryptedKey: eJytk9[...]oIJS6= Fields: {rjY=[...]KQ==, ZdWC[...]xg==, 6aLB[...]iw==, VD2h[...]6w==, 8UmX[...]MQ==}	rjY=[...]KQ==: JPAv[...]LA== ZdWC[...]xg==: w05J[...]Hg== 6aLB[...]iw==: 0wWu[...]vA== VD2h[...]6w==: eZu7[...]QQ== 8UmX[...]MQ==: sXaB[...]kQ==
National customs clearance	sender: 0x5F[...]FFE1 process_instance_id: 43175279 message_id: 64083548	EncryptedKey: eJytWU[...]gGl2Y= Fields: {fdoT[...]kA==, 2AkH[...]Rw==, 0bTn[...]6A==, RZVJ[...]rQ==, 4TXI[...]zw==}	fdoT[...]kA==: fUSZ[...]Bg== 2AkH[...]Rw==: dGp2[...]zA== 0bTn[...]6A==: TuR9[...]bA== RZVJ[...]rQ==: Pq8U[...]dQ== 4TXI[...]zw==: 0Izx[...]Mw==

(see step **0.4** above) and the *fdk* to the MA-ABE decryption algorithm running locally. (**2.8**) Mirroring the operations explained in step **2.1**, MA-ABE decrypts the symmetric key from the retrieved ciphertext. Only with the symmetric key, the Reader can obtain the original information artifact.

4.2 Data Structures

After the analysis of the software components and tasks employed in our approach, we focus on its core data structures: messages and policies.

Messages. Table 1 illustrates the messages we described in our running example in Sect. 2 along with a generated symmetric key for each message. Table 3 shows the messages as saved on the Data Store by the Data Owner after the encryption process explained in Sect. 4.1 (phase 2). Each file stored on the Data Store consists of one or more sections to be accessed by different actors (henceforth, *slices*). Every slice is

Table 4. Message policy examples

Message	Slice	Policy
Supply purchase order (ramp)		`43175279@2+ and (Manufacturer@1+ or (Supplier@1+ and International@1+))`
Export document	a	`Customs@A or (43175279@2+ and ((Supplier@1+ and International@1+) or Manufacturer@1+ or (Carrier@1+ and International@1+)))`
	b	`Customs@A or (43175279@2+ and (Supplier@1+ and International@1+))`
	c	`43175279@2+ and ((Supplier@2+ and International@1+) or Manufacturer@1+)`
	d	`Customs@A or (43175279@2+ and ((Supplier@1+ and International@1+) or Manufacturer@1+))`
National customs clearance		`Customs@A or (43175279@2+ and Manufacturer@1+)`

divided in three parts. **The metadata** contain the *message sender* (e.g., `0x82...1332` in Table 3), the *case id* (e.g., `43175279`), and the *message id* that uniquely identifies the message (e.g., `22063028`). **The body** is the encrypted information saved as key/-value entries (*fields*) for ease of indexation. For security, notice that neither the keys nor the values are in clear. **The header** consists of the *encrypted* symmetric *key* generated at step **2.1**, and the list of field keys that the body contains. In case two or more slices form the message (as in the case of the export document), each is marked with a unique *slice id* (e.g., `62618638`). We recall that a message is stored on the Data Store and retrievable through a hash, content-based resource locator. The resource locator can thus be attached to process execution data for monitoring and auditability purposes, in compliance with **R6**.

Policies. We use MA-ABE policies to specify read grants to message slices, thus enabling fine-grained access control as per **R1**. For example, the export document written by the international supplier of process instance `43175279` is partitioned in four slices as illustrated in Table 1. Table 4 shows the encoding of the policies that restrict access to specific classes of Readers, based on the attributes the Attribute Certifiers attested to in step **1.1**.

Henceforth, we will use the following notation to encode a policy P. We shall use $\mathrm{Attr}@X$ as a shorthand notation for a policy indicating that an authority Auth (if X is Auth) or *at least* $n \geqslant 1$ authorities (if X is n+) generate the key based on the verification of attribute Attr. Compound policies can be formed by joining $\mathrm{Attr}@X$ propositions with `or` and `and` logical operators. For instance, `(Customs@A or Supplier@1+)` declares that only authority `A` can authorize customs, whereas *any* authority can generate the *dk* for suppliers, and that only customs or suppliers can read a message. To sum up, we will henceforth use the following grammar for policies P:

$$P ::= \mathrm{Attr}@X \quad | \quad P \texttt{ and } P' \quad | \quad P \texttt{ or } P' \quad \textit{where } \mathrm{Attr} \textit{ is an attribute;}$$
$$X ::= \mathrm{Auth} \quad | \quad n\texttt{+} \quad \textit{where } \mathrm{Auth} \textit{ is an authority and } n \textit{ a positive integer.}$$

Notice that we enable the selection of a specific *dk* forger for backward compatibility towards single-authority frameworks. The downsides are that *(i)* no key is generated if that authority is down (if `A` crashed, e.g., a user cannot be recognized as a customs body), and *(ii)* a corrupted authority could take over the generation of an *fdk* if only one attestation is necessary (theirs). Therefore, special attention must be paid in the writing of policies. `43175279@2+` requires that *at least* two authorities attest to the participation of a user in case `43175279`. A user that is not authorized by all the required authorities cannot have the *fdk* as per the policy. Also, whenever multiple authorities are involved in the generation of the *fdk* by contributing to a part of it (the *dk*), only the user can compose the *fdk* and decrypt the ciphertext.

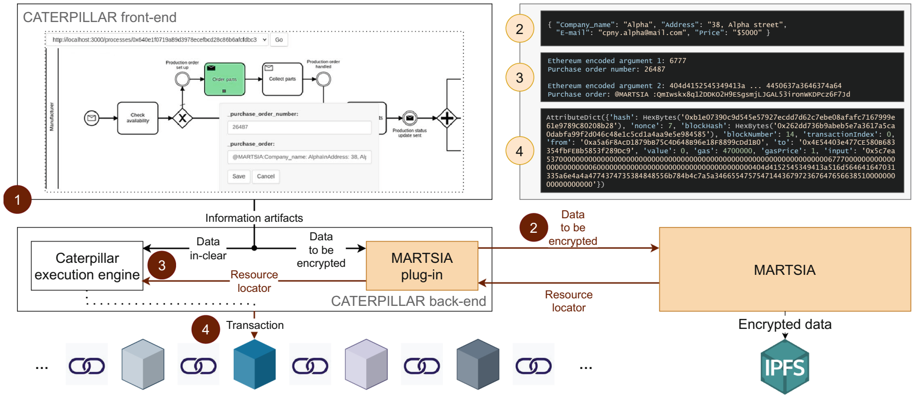


Fig. 5. A run of our integration of MARTSIA with Caterpillar [22]

In our example, the international shipment order is the first slice of the export document. It should be readable by the national and international customs, and by specific actors involved in the process instance: the sender (i.e., the international supplier), the manufacturer, and the international carrier. Additionally, we exert constraints on the authorities providing the *dk*: Customs are given the *dk* by Authority `A`, and at least two Authorities must declare that a Reader is involved in the given process instance. The other attributes can be attested to by any Authority. This composite rule translates to the following expression: `Customs@3+ or (43175279@2+ and ((Supplier@1+ and International@1+) or Manufacturer@1+ or (Carrier@1+ and International@1+)))`.

Thus far we have described the architecture of MARTSIA, along with its operations, employed techniques and data structures. Next, we focus on its realization and testing.

5 Implementation and Evaluation

MARTSIA is an approach aimed at securing the access to information at a fine-grained level in a distributed fashion. We have hitherto shown its security guarantees by design, using a multi-party process execution as a motivating scenario. In this section, we experimentally evaluate whether MARTSIA can deliver its guarantees and properties in a process context, and at what cost. The code of our prototype alongside the detailed results of our experiments can be found at github.com/apwbs/MARTSIA-Ethereum. We implemented the three contracts described in Sect. 4.1 as a single instance in Solidity, a programming language for the Ethereum Virtual Machine (EVM). We deployed the instance on the Sepolia (Ethereum), Mumbai (Polygon), and Fuji (Avalanche) testnets.[4] We created an IPFS local node(See footnote 2) to realize the Data Store, and used Python to encode the off-chain modules including the client-server communication channels.

[4] Goerli: sepolia.etherscan.io; Mumbai: mumbai.polygonscan.com; Avalanche: testnet.snowtrace.io. Accessed: 09 June 2023.

First, to demonstrate the adoption of MARTSIA as a secure data-flow management layer for process execution (thus meeting requirement **R6**), we created an integration module connecting our tool with a state-of-the-art blockchain-based process execution engine tool, i.e., Caterpillar [22]. For our experiments, we used Caterpillar v.1.0.[5] The code to replicate and extend our test is publicly available in our repository.[6] As shown in Fig. 5, we insert a plug-in in the architecture of Caterpillar to use MARTSIA as an intermediate data layer manager, replacing the built-in data store for information securing. We illustrate our experiment with a simplified fragment of the running example (see Sect. 2) focusing on the purchase order sent from the manufacturer to the international supplier. The exchanged data artifact consists of a purchase order number, to be publicly stored on chain, and the confidential purchase order entries listed in the top row of Table 1. The user passes both the entries as input through the Caterpillar panel (see mark ① in Fig. 5), specifying the data they want to be secured by MARTSIA with a special prefix ("`@MARTSIA:`"). Our integration module captures the input and encrypts the indicated entry as explained in Sect. 4 so that only the supplier can read and interpret those pieces of information as per **R1** (②). Once the encryption is concluded, MARTSIA invokes the Caterpillar Smart Contract passing the first argument (the purchase order number) as is, and replaces the second argument with a marked IPFS link in place of the original data (③). The resource locator for the stored information is thus saved on the ledger by the process execution engine for future audits (④), yet not publicly readable (**R2**). Thereupon, the recipient of the confidential information (or the auditor, later on) can retrieve and decode the information with their secret key (**R4**) provided by the authority network (**R5**).

Aside from empirically showing the suitability of MARTSIA as a secure data-flow manager in an ensemble with a process execution engine, we remark that the cost overhead in terms of transaction fees required by MARTSIA is negligible with respect to the main process execution management. For example, the method running the activity `Order parts` of the BPMN in Fig. 1 on Caterpillar incurs 114494 gas units for execution with our inputs. The on-chain component of MARTSIA, detached from Caterpillar, requires 89772 gas units to store the IPFS link. As we use the second string input field of the Caterpillar's smart contract to save that resource locator, the separate gas consumption for MARTSIA is unnecessary and can be directly included in the overall process execution costs. Notice that the same activity execution saving the purchase order as plaintext (thus renouncing to the fine-grained confidentiality guarantees of MARTSIA) would have entailed a larger cost because the textual content is a longer string than an IPFS link: 116798 gas units. Auditability on process execution *and* secure message exchanges are thus guaranteed with low overhead, as stated in **R3**.

To gauge the gas expenditure and execution time of our system's on-chain components, we called the methods of the deployed Smart Contract daily on Sepolia, Fuji and Mumbai for 14 days (from 16 to 29 May 2023). All the experiment's transactions and gas measurements are available in our code repository.[7] The data we used to run the tests are taken from our running example (see Sect. 2). Table 5 illustrates the results.

[5] github.com/orlenyslp/Caterpillar. Accessed: 10 June 2023.
[6] github.com/apwbs/MARTSIA-Ethereum/tree/main/caterpillar-interaction.
[7] github.com/apwbs/MARTSIA-Ethereum/tree/main/tests. Accessed: 10 June 2023.

Table 5. Execution cost and timing of the steps that require an interaction with the blockchain

Platform	**Execution cost** [Gwei = ETH × 10^{-9}]				**Avg. latency** [ms]
	Contract deployment (1692955 gas units)	Steps **0.1** to **0.5** (476547 gas units)	Step **1.2** (67533 gas units)	Step **2.4** (89772 gas units)	
Sepolia (ETH)	2539432.514	714820.504	101299.501	134658.001	9288.574
Fuji (AVAX)	340498.771	95873.485	13586.538	18060.662	4278.099
Mumbai (MATIC)	1283.163	354.691	50.311	66.012	4944.807
	Off-chain execution time [ms]				
	0.000	2582.471	38.280	158.447	

We divide the measurement in four different phases: *(i)* the deployment of the Smart Contract; *(ii)* the initialization of the Authorities (steps **0.1** to **0.5**); *(iii)* the Reader certification (step **1.2**); *(iv)* the storage of a message by a Data Owner to save a message (step **2.4**). For all the above phases, the table shows the average gas consumed for execution (ranging from 67533 units for step **1.2** to 1692955 units for the Smart Contract deployment) and the cost converted in Gwei (i.e., 10^{-9} Ether). Along with the analysis of costs, we measured the time needed to perform the steps of our approach. Each step involves sending a transaction to the blockchain. Therefore, we separate the execution time between the off-chain data elaboration and the latency induced by the blockchain infrastructure. The average time required to store a transaction in a block ranges from approximately 4.3 sec (Fuji) to 9.3 sec (Sepolia). The off-chain passages require a lower time, from about 0.038 sec for the Reader certification (Step **1.2**) to the circa 2.6 sec needed for the cooperative work carried out by the Authorities during the initilization phase (steps **0.1** to **0.5**). More in-depth comparative analyses and a stress test of the architecture pave the path for future endeavors, as we discuss in Sect. 7 after a summary of the state of the art.

6 Related Work

In recent years, numerous approaches have been proposed to automate collaborative processes using blockchain technology [12] beyond the aforementioned Caterpillar [22]. Previous studies in the area have shown the effectiveness of blockchain-based solutions to add a layer of trust among actors in multi-party collaborations [40] even in adversarial settings [23], improve verifiability of workflows with model-driven approaches [22,37], allow for monitoring [13], mining [17], and auditing [10]. Interestingly, a more recent release of Caterpillar [21] enables the dynamic allocation of actors based on a language for policy bindings. MARTSIA has the capability to adjust roles dynamically as well, as access keys are created based on actors' attributes verified at runtime. These studies enhance the integration of blockchain technology with process management, unlocking security and traceability benefits. However, they primarily focus on the control-flow perspective and lack mechanisms for secure access control to data stored on public platforms. In contrast, our work focuses specifically on this aspect in the context of collaborative business processes and, as we demonstrated in Sect. 5, can complement existing blockchain-based process execution engines.

Another area of research related to our investigation is the protection of privacy and integrity of data stored on the blockchain. Several papers in the literature explore the use of encryption for this purpose. Next, we provide an overview of techniques. Hawk [18] is a decentralized system that utilizes user-defined private Smart Contracts to automatically implement cryptographic methods. Our approach does not require the encoding of custom smart contracts, as it is based on policies stored on chain to encrypt messages. Bin Li et al. [19] introduce RZKPB, a privacy protection system for shared economy based on blockchain technology. Similarly to MARTSIA, this approach does not involve third parties and resorts to external data stores. Differently from their approach, we link data on chain with the data stores so as to permanently store the resource locators. Henry et al. [16] employ smart contracts that handle payment tokens. Banks operate as trustworthy intermediaries to preserve privacy. MARTSIA pursues confidentiality of exchanged information too, although it does not resort to central authorities (the banks) to this end. Rahulamathavan et al. [32] propose a new blockchain architecture for IoT applications that preserves privacy through the use of ABE. We also utilize ABE in our approach, but MARTSIA integrates with existing technologies, whilst their model aims to change the blockchain protocol. Benhamouda et al. [3] introduce a solution that enables a public blockchain to serve as a storage place for confidential data. As in our approach, they utilize shared secrets among components. However, their approach discloses the secret when determined conditions are fulfilled, whereas MARTSIA does not reveal secret data on the blockchain. In the healthcare domain, Wang et al. [39] create a secure electronic health records system that combines Attribute-Based Encryption, Identity-Based Encryption, and Identity-Based Signature with blockchain technology. Their architecture is different from ours as the hospital has control over the patient's data and sets the policies, whereas solely the data owners manage data in MARTSIA. Tran et al. [38] and Pournaghi et al. [31] propose approaches for decentralized storage and sharing based on private blockchains. We operate in the context of public blockchains to leverage the higher degree of security given by the general validation of transactions. Athanere et al. [1] present an approach where the data owner encrypts the file, and then a hashed version of it is stored on a cloud server. The data owner encrypts the data with the public key of the message reader, and the necessary public parameters are generated by an administrator. MARTSIA differs from this solution because it uses MA-ABE and symmetric key encryption to encrypt the data instead of a public key. Pham et al. [30] propose an idea for a decentralized storage system named B-Box, based on IPFS, MA-ABE and blockchains. Though we resort to those building blocks too, we include mechanisms for secure initialization of the authority network, allow for fine-grained access control on message parts, and impede by design any actor from accessing keys.

7 Conclusion and Future Remarks

In this work, we introduce MARTSIA, a technique that merges blockchain technology with Multi-AuthorityAttribute-Based Encryption (MA-ABE) to regulate data access in the scenario of multi-party business operations. Additionally, our method employs IPFS for preserving information artifacts, access regulations, and metadata. We utilize smart contracts to keep the user attributes, establish the access grants to the process participants, and save the connection to IPFS files. MARTSIA allows for a detailed specification of access permissions, ensuring data reliability, persistence, and irrefutability, thus enabling auditability with minimal added costs.

Our approach exhibits limitations we aim to overcome in future work. If a Data Owner wants to revoke access to data for a particular Reader, e.g., they can change the policy and encrypt the messages again. However, the old data on IPFS would still be accessible. Therefore, we are considering the usage of InterPlanetary Name System (IPNS), as it allows for the replacement of existing files. With it, a message can be replaced with a new encryption thereof that impedes Readers whose grant was revoked to access it. More generally, the life-cycle of data artifacts, policies and smart contracts constitutes a management aspect worth investigating. From a technological perspective, we are working on the implementation of MARTSIA on other public blockchain platforms such as Algorand[8] [7] to analyze the benefits and challenges stemming from different DLTs, including costs. Also, we are developing an alternative key request protocol for readers that adopts the blockchain as a communication layer so as to avoid direct channels between readers and authorities. In light of the considerable impact that a correct expression of policies has on the overall approach, we envision automated verification and simulation of policies for future work to properly assist the users in their policy specification task. Future endeavors also include the integration of Zero Knowledge Proofs [15] with ABE to yield better confidentiality and privacy guarantees, and of decentralized identifiers [29] and oracles [2] to verify data ownership. We plan to conduct a formal threat analysis to prove the security of our approach, and run field tests for the empirical evaluation of its robustness.

Acknowledgements. The work of E. Marangone and C. Di Ciccio was partially funded by the Cyber 4.0 project BRIE, the Sapienza project DRONES, by project SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU, and by project PINPOINT (B87G22000450001) under the PRIN MUR program.

References

1. Athanere, S., Thakur, R.: Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing. J. King Saud Univ. - Comput. Inf. **34**(4), 1523–1534 (2022)
2. Basile, D., Goretti, V., Di Ciccio, C., Kirrane, S.: Enhancing blockchain-based processes with decentralized oracles. In: González Enríquez, J., Debois, S., Fettke, P., Plebani, P., van de Weerd, I., Weber, I. (eds.) BPM 2021. LNBIP, vol. 428, pp. 102–118. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85867-4_8
3. Benhamouda, F., et al.: Can a public blockchain keep a secret? In: Pass, R., Pietrzak, K. (eds.) TCC 2020. LNCS, vol. 12550, pp. 260–290. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64375-1_10
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: SP, pp. 321–334 (2007)
5. Blum, M.: Coin flipping by telephone. In: CRYPTO, pp. 11–15 (1981)

[8] A preliminary version is available at github.com/apwbs/MARTSIA-Algorand.

6. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_28
7. Chen, J., Micali, S.: Algorand: a secure and efficient distributed ledger. Theor. Comput. Sci. **777**, 155–183 (2019)
8. Chi, A., Yao, C.: Protocols for secure computations (extended abstract). In: FOCS, pp. 160–164 (1982)
9. Chi, A., Yao, C.: How to generate and exchange secrets (extended abstract). In: FOCS, pp. 162–167 (1986)
10. Corradini, F., Marcelletti, A., Morichetta, A., et al.: Engineering trustable and auditable choreography-based systems using blockchain. ACM Trans. Manage. Inf. Syst. **13**(3) (2022)
11. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2003)
12. Di Ciccio, C., Cecconi, A., Dumas, M., García-Bañuelos, L., et al.: Blockchain support for collaborative business processes. Inform. Spektrum **42**, 182–190 (2019)
13. Di Ciccio, C., Meroni, G., Plebani, P.: On the adoption of blockchain for business process monitoring. SoSyM **21**(3), 915–937 (2022)
14. Feng, Q., He, D., Zeadally, S., Khan, M.K., Kumar, N.: A survey on privacy protection in blockchain system. J. Netw. Comput. Appl. **126**, 45–58 (2019)
15. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Providing Sound Foundations for Cryptography, pp. 203–225 (2019)
16. Henry, T., Hatin, J., Kazmierczak, L., Laga, N., Gaaloul, W., Bertin, E.: Random-value payment tokens for on-chain privacy-preserving payments. In: CoopIS, pp. 223–241 (2022)
17. Klinkmüller, C., Ponomarev, A., Tran, A.B., Weber, I., van der Aalst, W.M.P.: Mining blockchain processes: extracting process mining data from blockchain applications. In: BPM Blockchain and CEE Forum, pp. 71–86 (2019)
18. Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: SP, pp. 839–858 (2016)
19. Li, B., Wang, Y.: RZKPB: a privacy-preserving blockchain-based fair transaction method for sharing economy. In: TrustCom/BigDataSE, pp. 1164–1169 (2018)
20. Liu, Z., Jiang, Z.L., Wang, X., et al.: Multi-authority ciphertext policy attribute-based encryption scheme on ideal lattices. In: ISPA/IUCC/BDCloud/SocialCom/SustainCom, pp. 1003–1008 (2018)
21. López-Pintado, O., Dumas, M., García-Bañuelos, L., Weber, I.: Controlled flexibility in blockchain-based collaborative business processes. Inf. Syst. **104**, 101622 (2022)
22. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., Ponomarev, A.: CATERPILLAR: a business process execution engine on the Ethereum blockchain. Softw., Pract. Exper. **49**(7), 1162–1193 (2019)
23. Madsen, M.F., Gaub, M., Høgnason, T., et al.: Collaboration among adversaries: distributed workflow execution on a blockchain. In: FAB, pp. 8–15 (2018)
24. Marangone, E., Di Ciccio, C., Weber, I.: Fine-grained data access control for collaborative process execution on blockchain. In: BPM Blockchain and RPA Forum, pp. 51–67 (2022)
25. Mendling, J., Weber, I., et al.: Blockchains for business process management - challenges and opportunities. ACM Trans. Manage. Inf. Syst. **9**(1), 4:1–4:16 (2018)
26. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_31
27. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
28. National Institute of Standards and Technology: Announcing the Advanced Encryption Standard (AES). Technical report, Federal Information Processing Standards Publication 197 (2001)

29. Norta, A., Matulevicius, R., Leiding, B.: Safeguarding a formalized blockchain-enabled identity-authentication protocol by applying security risk-oriented patterns. Comput. Secur. **86**, 253–269 (2019)
30. Pham, V.D., Tran, C.T., Nguyen, T., Nguyen, T.T., et al.: B-Box - a decentralized storage system using IPFS, attributed-based encryption, and blockchain. In: RIVF, pp. 1–6 (2020)
31. Pournaghi, S., Bayat, M., Farjami, Y.: MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption. JAIHC **11** (2020)
32. Rahulamathavan, Y., Phan, R.C.W., Rajarajan, M., Misra, S., Kondoz, A.: Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. In: ANTS, pp. 1–6 (2017)
33. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems (reprint). Commun. ACM **26**(1), 96–99 (1983)
34. Rouselakis, Y., Waters, B.: Efficient statically-secure large-universe multi-authority attribute-based encryption. Cryptology ePrint Archive, Paper 2015/016 (2015)
35. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
36. Stiehle, F., Weber, I.: Blockchain for business process enactment: a taxonomy and systematic literature review. In: BPM Blockchain and RPA Forum, pp. 5–20 (2022). https://doi.org/10.1007/978-3-031-16168-1_1
37. Tran, A.B., Lu, Q., Weber, I.: Lorikeet: a model-driven engineering tool for blockchain-based business process execution and asset management. In: BPM Demos, pp. 56–60 (2018)
38. Tran, C.T., Pham, V.D., Nguyen, T., Dinh, H.H.Q., et al.: A novel approach for developing decentralized storage and sharing systems. In: BSCI, pp. 85–90 (2021)
39. Wang, H., Song, Y.: Secure cloud-based EHR system using attribute-based cryptosystem and blockchain. J. Med. Syst. **42**(8), 152 (2018)
40. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted business process monitoring and execution using blockchain. In: BPM, pp. 329–347 (2016)
41. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger (2014)
42. Xu, X., Weber, I., Staples, M.: Architecture for Blockchain Applications. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-03035-3
43. Zhang, R., Xue, R., Liu, L.: Security and privacy on blockchain. ACM Comput. Surv. **52**(3) (2019)

Model-Based Software Engineering

Building an Ontological Bridge Between Supply Chain Resilience and IoT Applications

Martijn Koot(✉), Martijn R. K. Mes, and Maria E. Iacob

University of Twente, Drienerlolaan 5, 7581WT Enschede, The Netherlands
{m.koot,m.r.k.mes,m.e.iacob}@utwente.nl

Abstract. The complexity of modern-day supply chains makes logistics operations more vulnerable towards disturbances, which endangers sustainability goals in the short-term. Local disturbances might effect logistics at large, as we typically see in congested urban areas. As a consequence, the Internet of Things (IoT) is gaining attention as a novel paradigm that promotes interconnected networks of context-aware electronic devices used for remote monitoring and control. These capabilities may stimulate anticipatory behaviour and more resilient supply chains, but a clear framework prescribing which objects to empower with electronic devices is still lacking. In this paper, we aim to semantically bridge the resilience and IoT paradigms in logistics environments. The ontology is developed by means of a bibliometric- and systematic literature study in search of essential concepts, and a field study to evaluate the ontology's effectiveness. Our ontology can form the basis to enhance resilience by replacing risk assessments with condition-based control mechanisms, resulting in better cooperation between human and software agents to resolve disturbances quicker, and more accurate training of machine learning algorithms in favour of autonomous decision making.

Keywords: Resilience · Internet of Things · Supply Chain Management · Logistics · Ontology

1 Introduction

In response to the increasing vulnerability of today's supply chains, logistics organizations aim to prepare themselves proactively towards potential threats [50]. Recent incidents highlight the necessity to build-in capacity to absorb disturbances and respond adequately. For example, temporary capacity outages (e.g., natural disasters, Suez canal blockage, or Covid-19 lockdowns) cause severe delivery delays throughout supply chains due to suspended manufacturing and logistics activities, and widespread congestion at logistics hubs once activities

This work is supported by the Netherlands Organization for Scientific Research (NWO) [grant number 628.0098.015].

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 79–96, 2024.
https://doi.org/10.1007/978-3-031-46587-1_5

restart again [19,34,38]. The large-scale impact of similar one-off events explains why the concept of *Supply Chain Resilience* (SCR) became well accepted among business practitioners and academics. However, some disruptions develop rather slowly, or gradually emerge from a collection of smaller events such as traffic jams, lost packages, or no-shows. At first glance, these minor disruptions may not seem worthwhile resolving, but together they still may cause severe functionality losses.

Luckily, advances in remote monitoring technologies have raised awareness among logistics planners regarding their operational performances in real-time [35]. As a result of various hardware-driven advancements, modern-day fleets are equipped with a variety of sensing, actuating, or identification devices (e.g., spatially distributed sensors, GPS receivers, RFID tags, or mobile devices), and continuously transmit the gathered data towards a centralized Fleet Management System [18]. The installation of remote sensory, communication, and data processing technologies resulted in an integrated network of uniquely addressable electronic devices better known as the *Internet of Things* (IoT) [7], a research paradigm that might extend the concept of physical monitoring with autonomous learning and decision making [40].

The design of a resilient system can be challenging due to the unpredictable and uncertain conditions of unforeseen failure modes [67]. Real-time monitoring can help logistics planners to observe their operations and, if needed, signal an alarm once a system change or failure is detected [52]. Therefore, we hypothesize that the installation of remote monitoring and distributed computing devices can provide better visibility, awareness and learning capabilities throughout supply chains [15,46,64], allowing disruptions to be anticipated, minimized, or avoided [67]. While plenty of ontologies exist for both paradigms separately, only a few articles have been found aiming to improve supply chain resilience by means of IoT advancements [3]. This lack of integration results in hardware-driven IoT implementations on the one hand [18], and hypothetical risk assessments on the other, without any clear instructions how to speed up recovery processes by means of condition-based monitoring. Therefore, the question we aim to address in this paper is how logistics planners can better maintain performances in the presence of disturbances by relying on integrated networks of context-aware resources, which demonstrates the need to construct a semantic model that integrates both resilience and IoT paradigms at the most fundamental level.

We answer the above research question by applying a structured ontological development approach [23], including a bibliometric- and systematic literature study in search of key objects, associations, and attributes. The envisioned ontology enables us to bridge the resilience and IoT paradigms in logistics environments, and should support enhanced visibility and awareness of logistics operations as well as provide a basis to create data for both autonomous learning and decision making. In Sect. 2, we provide a brief theoretical background on both the resilience and IoT paradigms. Our methodology in Sect. 3 explains how we integrate our bibliometric- and systematic literature search results into the development of an ontological framework, while the results themselves are discussed in Sect. 4. In Sect. 5, we demonstrate the usefulness of this research by mapping a use case with the aid of our ontological framework, while the conclusions and further research directions are discussed in Sect. 6.

2 Theoretical Background

Although the term resilience is used in a variety of disciplines, the concept always relates to an object's ability to effectively respond to disruptions and return to a more stable or desirable state [12]. In other words, resilience is associated with the capacities needed to recover from performance losses due to system changes [67]. Resilient behaviour is often depicted by means of a resilience curve, where system performances $P(t)$ are plotted against time t to recognize the alternative states that organizations have to pass through in response to disruptions. A resilient system is therefore not insensitive towards changing environments, as you might expect from the closely related term "robustness", but expects that disruptions might occur and should be approached proactively to withstand them [43]. To reduce risks, system engineers have to think in advance how to incorporate event readiness, provide an efficient and effective response, and be capable of recovering to obtain more desirable performances again [43,50].

In our research, the system of interest refers to modern-day supply chains, which are characterized by complex networks that gradually convert raw materials into finished products and services, including raw material extractors, manufacturing facilities, warehouses, distribution centers and retailers [26]. In this network, logistics operations are used to meet customers' requirements by orchestrating both forward and reverse movements of goods, services, and related information in between the point-of-origin and the point-of-destination [37]. Physical movements in supply chains could either represent a simple transportation line (e.g., a truck, train, or ship), or consist of a multi-modal network involving two or more transportation modes, transfer facilities, and logistics operators [26]. Therefore, resilient logistics systems are characterized by on-time pickups and/or deliveries, which requires carriers to reschedule operations when disturbances threaten the agreements made.

Over the years, various ontologies have been proposed to semantically describe how supply chain actors allocate physical resources (e.g., buildings, machinery, material handling equipment, load carriers, IT hardware, and employees) to support their value-adding activities (e.g., [24,28,44,65,68]). The majority of these ontologies either use the Enterprise Ontology [63] or the Supply Chain Operations Reference Model (SCOR) [6] as their foundation [30,57]. Some of the supply chain ontologies focus on logistics operations specifically (e.g., [5,56]), but the number of semantic models supporting the concept of resilience is relatively scarce. Comparable semantic models can be found for ontologies related to risk management (e.g., [29,48,59]). Most of these risk-oriented ontologies share the objective to mitigate performance losses by defining all potential threats in advance, and monitor operations and various information sources in search for those threats accordingly.

The IoT paradigm aims to equip daily physical objects with remote sensory, communication, and data processing technologies, resulting in an integrated network of uniquely addressable electronic devices [7,40]. Integrating the virtual and physical world by means of the latest IT advancements will create more intelligent services for two reasons [16,46]. First, sensing, actuating, and identification devices can be used to provide better visibility among complex systems by observing an object's status and its surroundings in real-time [35,64]. Second, the internet-based network of electronic devices provides a solid foundation for distributed computing systems [15,46], which in turn fosters collaborative communication and learning among context-aware objects to autonomously anticipate on changing environments [7,16,40]. The dynamic, heterogeneous, and object-oriented nature of IoT networks requires a flexible design, which is commonly achieved by applying a multi-layered Service Oriented Architecture (SOA), to guarantee scalability, interoperability, and modularity [2,16,40]. The architecture's dimensions may differ for each IoT application, but consists at least of the following layers [2,40,64]:

– **Perception layer**, physical hardware used for real-time data gathering, resource identification, or process control (e.g., wireless sensors/actuators, RFID tags, geo-spatial location receivers, or mobile devices);
– **Network layer**, middleware technologies that connect, coordinate and share data among physical objects and computational nodes (e.g., WSN, cellular networks, internet, Wi-Fi, or other wired/wireless network protocols);
– **Application layer**, software and business logic working together to make the IoT's functionalities available to the end-user by means of services.

Over the years, various ontologies have been proposed to enhance interoperability among the heterogeneous IoT devices [8,16,46,47]. Since remote monitoring is one of the IoT's unique capabilities, most IoT ontologies use the Semantic Sensor Network (SSN) standard proposed by [17] to describe sensing resources in terms of capabilities, measurement processes, observations and deployments. While the SSN ontology has become one of the most well known standards in the IoT domain, other ontologies have been proposed to overcome its shortcomings [8], for example:

– **IoT-Lite**: a lightweight version of the SSN ontology, aiming to reduce complexity and semantic processing time for real-time sensor discovery [11];
– **IoT-O**: a modular IoT ontology that semantically describes how to make IoT systems aware of their environment by means of connected devices (e.g., sensors, or actuators), services, and lifecycle and energy management [58];
– **OntoSensor**: a detailed ontology focusing on sensor specifications, including their properties, capabilities, and services [55];
– **FIESTA-IoT**: an attempt to unify the IoT's best practices into one semantic model that can address interoperability by focusing on the underlying testbeds' resources and associated observations [1].

The IoT ontologies listed above simply represent a snapshot from all semantic models available, because plenty of heterogeneity problems exist in IoT research

[8]. However, the IoT's main contribution is to extend the primary functionalities of physical objects with IT-based services, which can support organizational decision making. Therefore, any IoT ontology includes at least one of the elementary classes related to physical objects, IT components (hardware and software), and delivered services.

3 Research Methodology

The risk-oriented ontologies discussed in Sect. 2 aim to monitor operations in search for a predetermined set of threats, while the perception-oriented IoT ontologies focus more on interoperability among heterogeneous devices. Although it makes sense that IoT devices can enhance supply chain resilience thanks to their remote monitoring and distributed computing capabilities, it remains nontrivial how to achieve this. Semantic modelling could be helpful in this, but there are only a limited number of ontologies available that aim to improve supply chain resilience by means of IT advancements [3]. In this section, we introduce the overall procedure to construct our own ontology bridging both paradigms, including a description of our bibliometric- and systematic literature studies. We use the research methodology proposed by [23], called *Methontology*, to conceptualize the objects, relationships and attributes of our semantic model, because this approach aligns with our objective to integrate existing ontologies as much as possible. The Methontology proposes a structured method for building ontologies, where developers have to go through six interrelated states of the ontology life cycle. Due to the exploratory nature of this research, we focus on the first four developmental states before proceeding with both the implementation and maintenance states:

1. **Specification**: explain the ontology's purpose, its intended uses, and scope.
2. **Conceptualization**: structure the acquired knowledge into a conceptual model, including key concepts, relationships and attributes.
3. **Formalization**: transform the conceptual model into formal descriptions.
4. **Integration**: guarantee re-usability by relying on existing ontologies.

During the ontology development process, we carry out three supportive activities as well. First, we acquire more knowledge by performing an in-depth literature study, consisting of both bibliometric- and systematic approaches, to conceptualize the ontology's objects, relationships and attributes. Second, we document our formal ontology in Sect. 4 by means of a class diagram according to the Unified Modelling Language (UML), version 2.5.1. (https://www.omg.org/spec/UML). Note that UML supports our design of IoT-based and resilient logistics systems, but other languages can assist our semantic modeling as well, like OntoUML [31], or OWL [65]. Finally, we evaluate the usefulness of our ontology by mapping a real-life use case in Sect. 5.

By studying common groupings of keywords published in relevant literature, we gain an understanding of how emerging IoT technologies can improve supply chain resilience. Therefore, we initiate our research by means of two independent

Table 1. Inclusion and exclusion criteria used for both bibliometric- and systematic literature searches.

Criterion type	Criterion ID	Definition
Inclusion (Screening)	Criterion 1	All literature should be written in English, and the source should either be fully published, or in press
	Criterion 2	Include academic publications only (e.g., articles, conference papers, reviews, books, or book chapters)
	Criterion 3	The publications' subject areas have to align with the logistics domain (e.g., Decision Sciences, Business Management, or Economics)
Inclusion (Content)	Criterion 4	The publication should propose a unique and self-developed ontology related to IoT or Supply Chain Resilience
	Criterion 5	The publication should explicitly state the logistics activities that it intends to describe semantically
Exclusion	Criterion 6	Remove duplicate articles originating from the same research project, only save the most recent version for further reading
	Criterion 7	Remove articles whose full text is not fully accessible

bibliometric searches in *Scopus* to find relevant keywords associated with either resilience or IoT applications in logistics environments.

Bibliometric search queries:

1. (Resilien*) AND ("Supply chain" OR Logistics OR Transport*)
2. ("Internet of Things" OR IoT) AND ("Supply chain" OR Logistics OR Transport*)

Both bibliometric search queries are seeking for matches within either the title, abstract, or keywords, and are bounded by the inclusion criteria from Table 1. For the remaining search results, we export a RIS-file to visualize the 30 most frequently co-occurring keywords by means of a bibliometric network using the software *VOSviewer* (https://www.vosviewer.com). VOSviewer automatically emphasizes the most frequent keyword sets and searches for appropriate clusters based on the keywords' association strength. General keywords related to research type are removed from the bibliometric network (e.g., literature study), and a thesaurus file is created to make sure that synonyms are not counted separately. Note that we use these common keywords to reshape the query for our systematic search approach.

After our bibliometric search approach, we also search for similar resilience and IoT ontologies by means of a systematic literature study. Once again, we initiate our search by creating two independent queries in *Scopus*, using groupings of keywords resulting from our bibliometric search (see Sect. 4). Note that the resilience-related query includes additional keywords due to the lack of ontologies found that explicitly refer to supply chain resilience (see Sect. 2).

Systematic literature search queries:

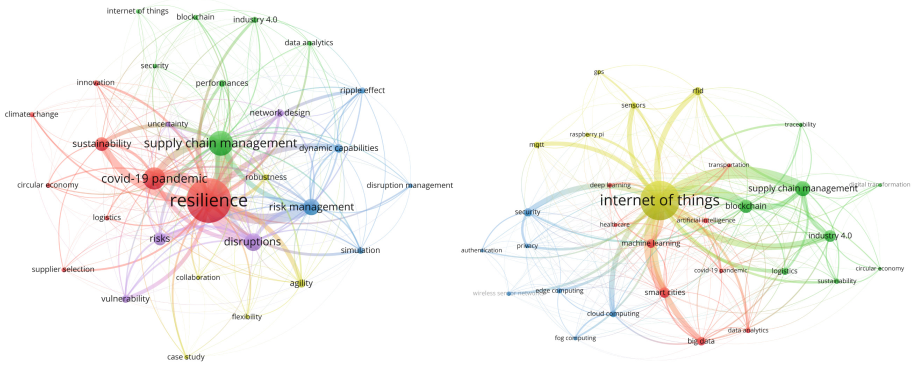
1. (Resilien* OR Disrupt* OR Disturb* OR Risk OR Vulnerab* OR Uncertain*) AND (ontolog*) AND ("Supply chain" OR Logistics OR Transport*)
2. ("Internet of Things" OR IoT) AND (ontolog*) AND ("Supply chain" OR Logistics OR Transport*)

Similar to the bibliometric search, we are seeking for matches within either the title, abstract, or keywords. We assess the relevance of the resulting publications by applying three layers of inclusion/exclusion criteria, as shown in Table 1. All remaining articles are fully read in search for essential building blocks of the IoT and resilience paradigms separately. In Sect. 4, we classify the semantic objects included in the SLR results.

4 Ontology Development

The ontological developments in this section are structured according to the specification, conceptualization, formalization, and integration states prescribed by the Methontology [23] in Sect. 3. The purpose of our ontology is to enable supply chains resolve disturbances more proactively by means of an integrated network of context-aware resources. Before physical resources can anticipate on system changes autonomously, we should clarify how to enhance visibility of logistics operations, and how to use sensory data to create a data set to train machine learning algorithms for autonomous decision making. The envisioned ontology should merge the resilience and IoT concepts to resolve disturbances more proactively, especially by empowering physical resources with remote sensory, communication and/or data processing devices. As a result, our ontology is best applied in the design of logistics networks, helping decision makers to simultaneously map out the network's risks, and determine which physical resources need to be equipped with uniquely addressable electronic devices. The ontology focuses on the physical resources (e.g., transportation modes, material handling equipment, and packaging material) required for the transportation of tangible goods in between two or more supply chain nodes.

In the conceptualization phase, we categorize the key constructs found during our literature study. Execution of the first bibliometric search query defined in Sect. 3 resulted in 2,827 publications related to supply chain resilience (search date: 5th of January 2023), including 6,080 different keywords once we have activated our thesaurus file. The 30 keywords most frequently associated with supply chain resilience are categorized in five clusters (see alternative markings in Fig. 1a). The resilience concept is strongly associated with the idea to build sustainable supply chains and circular economies (red cluster), which helps logistics planners to meet current demand efficiently without harming future generations. Recent disruptive events like the Covid-19 pandemic and climate change seem to encourage research towards supply chain resilience even more. The green cluster provides several data-driven paradigms that supply chain managers currently explore to maintain performances (e.g., Industry 4.0, Blockchain, Data analytics, and Internet of Things). The importance to build-in rescheduling capacity in advance is even better highlighted by the purple cluster, including keywords like disruptions, risks, vulnerability, and uncertainty. The blue cluster focuses more on the management techniques needed to tackle those threats from the purple cluster. Finally, researchers seem to be interested in both pro-active (e.g., agility and flexibility) and reactive approaches (e.g., robustness) to tackle disturbances, as can be seen by the yellow cluster.



(a) Resilience (cluster count = 5). (b) IoT (cluster count = 4).

Fig. 1. Co-occurrence networks of the 30 most frequently used keywords for (a) resilience and (b) IoT, both within logistics environments. (Color figure online)

Execution of the second bibliometric search query defined in Sect. 3 resulted in 2,608 publications regarding IoT applications in the logistics domain (search date: 10$^{\text{th}}$ of January 2023), including 6,340 different keywords once we have activated our thesaurus file. The IoT-related keywords are categorized in four clusters, as visualized by the alternative markings in Fig. 1b. The yellow cluster highlights the hardware-driven focus of the IoT paradigm due to the presence of various sensing an identification devices (sensors, GPS, RFID), communication protocols like Message Queuing Telemetry Transport (MQTT) and computation devices (Raspberry Pi). The blue cluster focuses more on the distributed computation architectures (e.g., edge, fog, and cloud computing), while the red cluster highlights the pattern searching paradigms more closely (e.g., big data, data analytics, machine learning, and artificial intelligence). Finally, the green cluster emphasizes the application of IT advancements in the logistic domain itself.

On the 22nd of March 2023, our systematic literature search defined in Sect. 3 resulted in 19 and 10 academic publications related to supply chain resilience and IoT applications, respectively, as visualized in the search roadmap in Fig. 2. We have read all 29 papers in search for the fundamental building blocks of

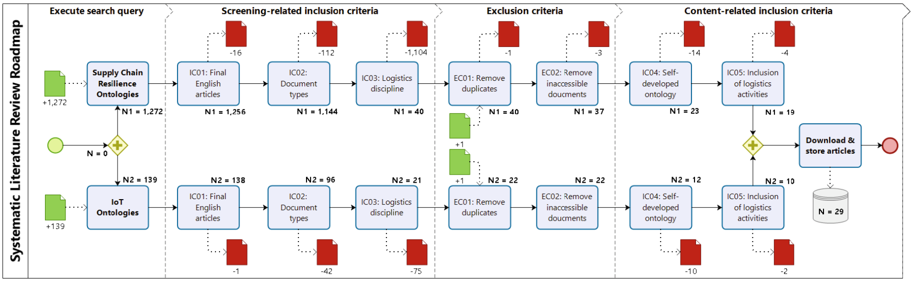


Fig. 2. Roadmap corresponding to the SLR strategies described in Sect. 3.

Table 2. Filled-in data extraction form, ranked in alphabetic order, highlighting the key concepts discussed in our systematic search results.

Object	Definition	Related terms	Resilience references	IoT references
Actions	Set of possible decisions an agent can make, given state of the corresponding object and/or system	Decision, Measure, Response	[4, 20, 21, 32, 45, 66, 69]	[36, 42, 51, 62]
Activity	Value-adding task related to the production or distribution of an organization's products and services	Process, Task	[4, 9, 10, 13, 20, 21, 27, 32, 53, 60, 61, 69]	[24, 62, 70]
Stakeholder	Person, department, or organization with a particular interest in the physical object	Actor, End-user, Operator	[4, 9, 10, 13, 14, 20, 21, 27, 39, 45, 53, 60, 61, 66]	[22, 24, 36, 42, 49, 62, 70]
Device	IT equipment needed for the acquisition, communication and processing of real-time data, including both hard- and software components	Actuator, Tag, Sensor, Supervision method	[20]	[22, 42, 62]
Disturbance	Event causing functionality losses throughout the logistics network	Alarm, Cause, Defect, Disruption, Event, Fault, Hazard, Perturbation, Resilience metric, Risk	[9, 10, 13, 14, 20, 32, 39, 45, 53, 54, 66, 69]	[22, 36, 42, 51, 62, 70]
Infrastructure	Public structures needed to facilitate value-adding activities that supply chain actors intend to perform (may refer facilitating IT architectures as well)	Connection (e.g., road, waterway, rail), Facility, Hub, Object, Platform, Static equipment	[10, 20, 21, 25, 41, 45, 60, 66]	[25, 36, 49]
Objective	The result aimed for by one or multiple stakeholders	Aim, Goal, Prediction, Target	[4, 9, 20, 32]	[42]
Performance	The observed state of physical objects in terms of its functional behaviour and/or surrounding circumstances	Condition, Context, Environment, KPI, Measurement, Observation, State, Status, Surroundings	[4, 9, 14, 20, 21, 25, 27, 32, 39, 60, 61]	[22, 25, 33, 36, 42, 51, 62, 70]
Procedure	A detailed description on how, when and where to execute a task, made in advance of the execution itself	Algorithm, Computational method, Governance, Planning, Policy, Schedule, Strategy	[4, 13, 21, 41, 60, 66]	[42, 62]
Product	Physical goods that organization intend to manufacture, consume, or transport	Container, Good, Material, Object, Package, Platform	[4, 9, 10, 27, 41, 45, 54, 61, 69]	[33]
Requirement	Necessary condition that physical objects should comply with for task executions	Constraint, Limitation, Quality of service, Specification	[10, 20, 41]	[42, 51, 62, 70]
Resource	Tangible asset that organizations need to support their primary activities	Asset, Object, Platform	[10, 14, 20, 21, 25, 27, 39, 41, 45, 60, 66, 69]	[22, 24, 25, 36, 42, 49, 51, 62, 70]
Service	Functionality that entities (e.g., activity, agent, or resource) can offer to their surroundings	Ability, Capability, Disposition, Function, Functionality	[4, 10, 13, 20, 60, 61, 69]	[42, 49, 51, 62, 70]
System	An interconnected network of infrastructural components	Network, Structure	[13, 54, 60, 61, 66]	[22, 49]

the corresponding ontologies, and marked all findings in our data extraction form in Table 2. The objects in Table 2 have been selected iteratively by comparing the semantic models found during our systematic literature search with the theoretical frameworks discussed earlier in Sect. 2. Note that we have provided a descriptive definition for all objects in Table 2, including a list of synonyms and/or strongly related terms. We will use the filled-in data extraction form and co-occurrence networks to formalize our ontology in the formalization phase.

The most frequent resilience-related keywords in Fig. 1a focus on dynamic capabilities to maintain supply chain performances in the presence of threatening disturbances. This risk-oriented view explains why most resilience ontologies in Sect. 2 and Table 2 extend their semantic models with entities like risks, events, disturbances, system/object requirements, resource conditions, or environmental statuses [48,59]. For example, multiple authors have defined a separate entity to represent risk sources (e.g., [9,10,13,14,36,39,45,66,69]), modelled disruptions and their causes explicitly (e.g., [53]), or quantified risks by examining the consequences of potential functionality losses (e.g., [32,54]). Since businesses orchestrate their activities by means of a planning process to fulfill objectives (e.g., [5,21,28,41,42,44,51,60,65]), most ontologies model the supply chain's value-adding activities and corresponding resource allocations first, and explicitly highlight the potential functionality losses second. This risk-oriented approach is less effective however when performances deteriorate in an unpredictable way due to a series of reinforcing events, especially when the disturbances' characteristics remain uncertain as well (e.g., frequency, time of introduction, impact area, disturbance duration, and severity). Therefore, we are in the need of a semantic model that prescribes stakeholders to remain resilient by rescheduling activities based on the performance losses predicted or observed, which is fulfilled by the GenCLOn ontology for city logistics [5]. We will use the vast majority of entities included in GenCLOn's top-level hierarchy as foundation for the resilience-oriented part of our ontology, including activities, Key Performance Indicators (KPIs), objectives, stakeholders, resources, and measures.

The most common IoT-related keywords in Fig. 1b emphasize the importance of artificial intelligence in logistics environments supported by remote sensing/identification devices cooperating in a distributed computation architecture. The object-oriented view of the IoT's perception layer enables logistics planners to automatically capture potential threats by observing the objects' features of interest in real-time (e.g., [11,17,20,22,42,62]), given that electronic devices are employed in such a way that operational performances can be monitored. Most supply chain ontologies use KPIs to track operational performances (e.g., [5,13,20,28,32,44,56,60,65,68]), while some IoT-focused papers refer to condition monitoring instead (e.g., [21,22,33,51]). A few authors formulate process requirements and/or constraints that need to be maintained in dynamic environments (e.g., [42,44,51,62,68]), which becomes more relevant when disruptions threaten the feasibility of the scheduled activities. Data-driven computational methods are required as well to transform observations of the system's state into effective measures (e.g., [17,20,42,51,62]). Therefore, we are in search of an IoT ontology prescribing similar mechanisms for both continuous monitoring and control (e.g., [17,58]). We will use the SSN ontology as foundation for the IoT-related entities of our ontology, because of its focus on "*...describing sensors and their observations, the involved procedures, the studied features of interest, ... , as well as actuators*" [17]. The SSN ontology is also regarded as foundation for most other IoT ontologies [8], which makes it easier to integrate our semantic model with other domain languages in later developmental stages.

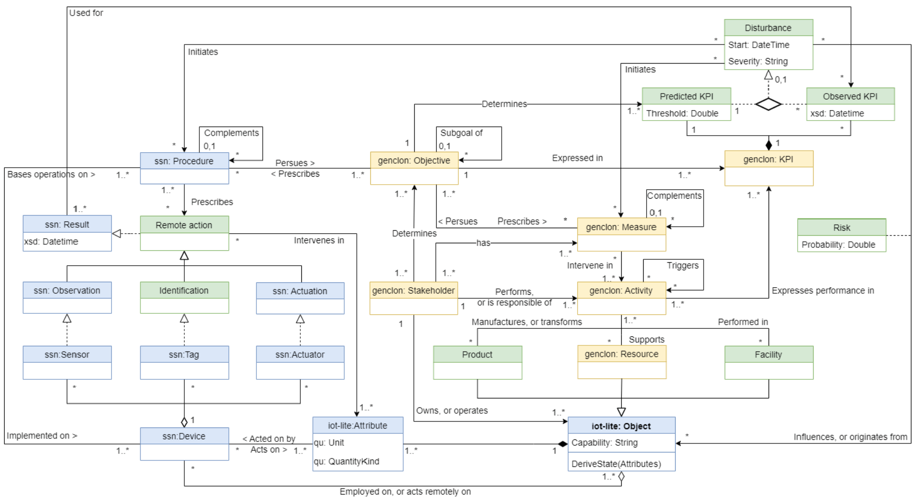


Fig. 3. Proposed ontology to bridge the resilience and IoT concepts in logistics environments (UML version 2.5.1). The resilience-related entities from the GenCLOn ontology [5] are marked yellow, while all IoT entities from both SSN [17] and IoT-Lite [11] ontologies are marked blue. Green marked entities are proposed by us, based on the SLR results from Table 2. (Color figure online)

By comparing both GenCLOn and SSN ontologies, as well as similar semantic models in Table 2, we can unambiguously observe distinctive modelling purposes for both paradigms based on the entities included (e.g., risk assessment of logistics resource allocations versus the empowerment of physical objects with electronic devices). Both paradigms share the need to control physical objects in dynamic environments however, since logistics flow can be disrupted once events change the objects' states over time [24]. Instead of focusing on the potential risks themselves, or the hardware's capabilities only, we advocate to constantly measure discrepancies between expected and observed performances, and register disturbances once threshold values are exceeded. Therefore, sensory, identification and/or actuation equipment should be employed in association with the objects' attributes used to quantify performances of the supported activity, as visualized in our ontology in Fig. 3. Actuators should be employed to intervene in objects' statuses once disturbances are either observed or predicted, and reschedule operations centrally when the objects' procedures are unable to recover from disruptions autonomously. The associations between KPIs, objects' characteristics, and condition-based actions showed up during our literature search a few times as well. For example, continuous monitoring can be used to: 1) both predict and measure the metrics of interest [4]; 2) detect trends, patterns and anomalies among the equipment's operational data [20]; and 3) orchestrate supply chain activities by using a more event-driven and context-aware approach [42].

5 Demonstration

We demonstrate the usefulness of our ontology by mapping a real-world IoT project on the entities included in Fig. 3, and assess to what extent our ontology can enhance resilience in similar IoT projects. We are particularly interested to what extent our semantic model can reflect reality in search for any incompleteness, inconsistencies, and redundancies. For this purpose, we conducted an observational study and informal interviews with a Dutch project consortium of manufacturers, IT providers, logistics carriers and hinterland container terminals, who envision a self-organizing logistics system. The project aims to employ LoRa sensors on trucks, barges, and containers, to empower those resources with local intelligence, which should result in more autonomy and dynamic rescheduling capabilities. The electronic devices can localize objects, observe their temperature, motion, and light in real-time, and raise notifications if needed by relying on WiFi and low energy Bluetooth technologies. Operational data is transferred to a centralized application every minute, where logistics planners use a dashboard to gain insights into real-time information on all transportation statuses.

While filling our semantic model with instances from this particular use case, it becomes clear that most stakeholders know how to employ their objects with electronic devices (blue entities in Fig. 3), enabling them to raise alarms when performances deteriorate. For example, the shipping containers represent the tangible objects equipped with LoRa devices, including multiple sensors and identification tags that continuously monitor the objects' physical attributes (e.g., location, temperature, humidity, motion, or light). The devices' procedures raise alarm when at least one observed physical attribute exceeds a pre-installed threshold value, which initiates human planners to think of corrective actions along the containers' logistical pathways (yellow entities in Fig. 3). For example, planners can switch transportation modalities when a container's estimated time of arrival is getting too close to it's due date as a result of unexpected events, which conflicts with the interests of the logistics service provider, the carrier, and customer. This alarm mechanism forms the essential link in between the IoT and supply chain resilience paradigms (green entities in Fig. 3), because it integrates the concepts of condition-based monitoring and corrective resource allocations from the SSN and GenCLOn ontologies, respectively. The objects' conditions are transformed in operational performance measurement by means of KPIs, which enables both logistics planners and the corresponding objects' procedures to mitigate disruptions as soon as performance discrepancies are observed, even without knowing the exact nature or cause of that event immediately (e.g., delays, shortages, breakdowns).

The validation exercise reveals that our ontology is capable of mapping a real-world IoT project intended to increase supply chain resilience. We demonstrated that the entities in Fig. 3 can be used to consistently describe the project's intentions without any redundancies. However, further development is needed to ensure the ontology's completeness as well. For example, our ontology mainly integrates the concept of remote condition monitoring and re-allocation of physical resources, while the required information systems are not represented directly yet (e.g., applications, dashboards, or external information sources). The validation exercise also clarifies the usefulness of our semantic model compared to ontologies originating from either IoT or resilience paradigms. For example, the project consortium envisions a self-organizing logistics system, but it remains unclear under which conditions rescheduling policies should intervene in the yellow entities of Fig. 3. Most threshold values relate to specific physical attributes only, which causes logistics planners to think of effective measures themselves every time more complex disruptions emerge. Our condition-based ontology could stimulate cooperation between humans and software agents to mitigate the negative consequences of disturbances quicker, and supports training machine learning algorithms for autonomous decision making utilizing the large data sets generated over time. The performance discrepancies can be used for supervised learning (e.g., use historical records to predict the emergence of new disruptions), unsupervised learning (e.g., cluster disturbance types based on the objects' characteristics), or reinforcement learning (e.g., prescribe objects to reallocate themselves based on the system's state). As an example of the latter, the authors from [25] propose an ontology to accurately describe the system's state, which supports agents to make better decisions by means of reinforcement learning, a key feature needed to move from descriptive- towards prescriptive analytics, i.e., more autonomous decision making.

6 Conclusion and Further Research

In this research, we aim to semantically bridge the concepts of supply chain resilience and IoT networks at the most fundamental level. Our semantic model integrates the SSN [17] and GenCLOn [5] ontologies and prescribes how to make logistics networks inherently more resilient by controlling resource allocations in real-time. Instead of focusing on the potential risks themselves, we advocate to shift attention to monitoring performance discrepancies, and give the physical objects more autonomy to recover from disturbances as quickly as possible. The benefits of this condition-based approach are twofold: 1) accelerated recovery operations due to improved collaboration between humans and software agents; and 2) the creation of data sets that can feed machine learning algorithms for more autonomous decision making. In future research, we aim to extend our ontology by evaluating semantic approaches from other disciplines, since the importance of resilience is recognized by various fields. For example, the idea of *Enterprise Resilience* focuses more on individual entities instead of GenCLOn's focus on networks, but could still enrich the concepts listed in Table 2 and Fig. 3. Our semantic associations are open to discussion as well. For example, in our ontology, rescheduling measures are based on the stakeholders' objectives, assuming that the planning details are prescribed by the SCOR model in the *activity* entity [6,28,68], while we could also advocate that operations are scheduled by means of strategies and planning processes instead of the overarching objectives directly, as prescribed by the IDEON ontology [44]. We also advocate exploring the ontology's completeness in more detail by means of more real-life case studies. Finally, we are eager to make our research of more practical use by implementing a web-based version of our ontology, by using well-known and compatible languages like OWL [65].

References

1. Agarwal, R., et al.: Unified IoT ontology to enable interoperability and federation of testbeds. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 70–75. IEEE (2016)
2. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of Things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tut. **17**(4), 2347–2376 (2015). https://doi.org/10.1109/COMST.2015.2444095
3. Al-Talib, M., Melhem, W.Y., Anosike, A.I., Garza Reyes, J.A., Nadeem, S.P., kumar, A.: Achieving resilience in the supply chain by applying IoT technology. Procedia CIRP **91**, 752–757 (2020). https://doi.org/10.1016/j.procir.2020.02.231
4. Ameri, F., Sormaz, D., Psarommatis, F., Kiritsis, D.: Industrial ontologies for interoperability in agile and resilient manufacturing. Int. J. Prod. Res. **60**(2), 420–441 (2022). https://doi.org/10.1080/00207543.2021.1987553
5. Anand, N., Yang, M., van Duin, J.H.R., Tavasszy, L.: GenCLOn: an ontology for city logistics. Exp. Syst. Appl. **39**(15), 11944–11960 (2012). https://doi.org/10.1016/j.eswa.2012.03.068
6. Association Supply Chain Management (ASCM): SCOR digital standard (2023). https://scor.ascm.org/. Accessed 24 Feb 2023
7. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010). https://doi.org/10.1016/j.comnet.2010.05.010
8. Bajaj, G., Agarwal, R., Singh, P., Georgantas, N., Issarny, V.: A study of existing ontologies in the IoT-domain. arXiv preprint arXiv:1707.00112 (2017)
9. Batlajery, B.V., Weal, M., Chapman, A., Moreau, L.: prFood: ontology principles for provenance and risk in the food domain. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC), pp. 17–24. IEEE (2018)
10. Benazzouz, T., Echchtabi, A., Charkaoui, A.: Ontology for risks in medicines supply chain: case of public hospitals in Morocco. MATEC Web Conf. **105**, 00012 (2017). https://doi.org/10.1051/matecconf/201710500012
11. Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K.: IoT-lite: a lightweight semantic model for the Internet of Things. In: 2016 INTL IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, pp. 90–97. IEEE (2016)
12. Bhamra, R., Dani, S., Burnard, K.: Resilience: the concept, a literature review and future directions. Int. J. Prod. Res. **49**(18), 5375–5393 (2011). https://doi.org/10.1080/00207543.2011.563826

13. Cao, S., Bryceson, K., Hine, D.: Improving supply chain risk visibility and communication with a multi-view risk ontology. Supply Chain Forum Int. J. **21**(1), 1–15 (2020). https://doi.org/10.1080/16258312.2020.1717990
14. Cao, T., Mu, W., Montarnal, A., Barthe-Delanoë, A.-M.: A method of ontology evolution and concept evaluation based on knowledge discovery in the heavy haul railway risk system. In: Camarinha-Matos, L.M., Afsarmanesh, H., Antonelli, D. (eds.) PRO-VE 2019. IAICT, vol. 568, pp. 220–233. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28464-0_20
15. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. IEEE IoT J. **3**(6), 854–864 (2016). https://doi.org/10.1109/JIOT.2016.2584538
16. Čolaković, A., Hadžialić, M.: Internet of Things (IoT): a review of enabling technologies, challenges, and open research issues. Comput. Netw. **144**, 17–39 (2018). https://doi.org/10.1016/j.comnet.2018.07.017
17. Compton, M.: The SSN ontology of the W3C semantic sensor network incubator group. J. Web Semant. **17**, 25–32 (2012). https://doi.org/10.1016/j.websem.2012.05.003
18. Crainic, T.G., Gendreau, M., Potvin, J.-Y.: Intelligent freight-transportation systems: assessment and the contribution of operations research. Transp. Res. Part C Emerg. Technol. **17**(6), 541–557 (2009). https://doi.org/10.1016/j.trc.2008.07.002
19. Cullinane, K., Haralambides, H.: Global trends in maritime and port economics: the COVID-19 pandemic and beyond. Marit. Econ. Logist. **23**(3), 369–380 (2021). https://doi.org/10.1057/s41278-021-00196-5
20. Dausch, M., Hsu, C.: Engineering service products: the case of mass-customising service agreements for heavy equipment industry. Int. J. Serv. Technol. Manage. **7**(1), 32–51 (2006)
21. Defèr, F., Schuh, G., Stich, V.: Towards a unified reliability-centered information logistics model for production assets. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds.) APMS 2020. IAICT, vol. 591, pp. 11–18. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57993-7_2
22. Feng, F., Pang, Y., Lodewijks, G.: Towards context-aware supervision for logistics asset management: concept design and system implementation. In: Ziemba, E. (ed.) AITM/ISM -2016. LNBIP, vol. 277, pp. 3–19. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53076-5_1
23. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering (1997)
24. Geerts, G.L., O'Leary, D.E.: A supply chain of things: the EAGLET ontology for highly visible supply chains. Decis. Support Syst. **63**, 3–22 (2014)
25. Ghanadbashi, S., Golpayegani, F.: Using ontology to guide reinforcement learning agents in unseen situations. Appl. Intell. **52**(2), 1808–1824 (2021). https://doi.org/10.1007/s10489-021-02449-5
26. Ghiani, G., Laporte, G., Musmanno, R.: Introduction to Logistics Systems Planning and Control. Wiley, Hoboken (2005). https://doi.org/10.1002/0470014040
27. Gliem, D., Jessen, U., Wenzel, S., Kusturica, W., Laroque, C.: Ontology-based forecast of the duration of logistics processes in one-of-a-kind production in SME. Logist. Res. **15**(1), 5 (2022)
28. Gonnet, S., Vegetti, M., Leone, H., Henning, G.: Scontology: a formal approach toward a unified and integrated view of the supply chain. In: Cruz-Cunha, M.M., Cortes, B.C., Putnik, G.D. (eds.) Adaptive Technologies and Business Integration: Social, Managerial and Organizational Dimension, pp. 137–158. IGI Global, Hershey, Pennsylvania, USA (2007)

29. Grandry, E., Feltus, C., Dubois, E.: Conceptual integration of enterprise architecture management and security risk management. In: 2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops, pp. 114–123. IEEE (2013)
30. Grubic, T., Fan, I.-S.: Supply chain ontology: review, analysis and synthesis. Comput. Ind. **61**(8), 776–786 (2010)
31. Guizzardi, G.: Ontological foundations for structural conceptual models. Ph.D. thesis, University of Twente, October 2005
32. Hachicha, M., Fahad, M., Moalla, N., Ouzrout, Y.: Performance assessment architecture for collaborative business processes in BPM-SOA-based environment. Data Knowl. Eng. **105**, 73–89 (2016)
33. Hu, S., Wang, H., She, C., Wang, J.: AgOnt: ontology for agriculture Internet of Things. In: Li, D., Liu, Y., Chen, Y. (eds.) CCTA 2010. IAICT, vol. 344, pp. 131–137. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18333-1_18
34. Inoue, H., Todo, Y.: Firm-level propagation of shocks through supply-chain networks. Nat. Sustain. **2**(9), 841–847 (2019)
35. Ivanov, D., Dolgui, A.: A digital supply chain twin for managing the disruption risks and resilience in the era of industry 4.0. Prod. Plan. Control **32**(9), 775–788 (2021)
36. Khezaz, A., Hina, M.D., Guan, H., Ramdane-Cherif, A.: Driving context detection and validation using knowledge-based reasoning. In: KEOD, pp. 219–226 (2020)
37. Lambert, D.M.: Supply chain management, chap. 1. In: Lambert, D.M. (ed.) Supply Chain Management: Processes, Partnerships, Performance, pp. 1–22. Supply Chain Management Institute, Sarasota (2008)
38. Lee, J.M., Wong, E.Y.: Suez Canal blockage: an analysis of legal impact, risks and liabilities to the global supply chain. In: MATEC Web of Conferences, vol. 339. EDP Sciences (2021)
39. Li, J., Gou, J., Mu, W., Peng, L.: Modeling of railway risk inter-relation based on the study of accident context. In: Tina Comes, F.B., Hanachi, C., Lauras, M., Montarnal, A. (eds.) Proceedings of the 14th International Conference on Information Systems for Crisis Response And Management, pp. 328–340. Iscram, Albi (2017)
40. Li, S., Xu, L.D., Zhao, S.: The Internet of Things: a survey. Inf. Syst. Front. **17**(2), 243–259 (2015)
41. Liu, Y., et al.: Digital twin-driven approach for smart city logistics: the case of freight parking management. In: Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., Romero, D. (eds.) APMS 2021. IAICT, vol. 633, pp. 237–246. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85910-7_25
42. Lyu, M., Biennier, F., Ghodous, P.: Integration of ontologies to support control as a service in an industry 4.0 context. Serv. Oriented Comput. Appl. **15**(2), 127–140 (2021)
43. Madni, A.M., Jackson, S.: Towards a conceptual framework for resilience engineering. IEEE Syst. J. **3**(2), 181–191 (2009)
44. Madni, A.M., Lin, W., Madni, C.C.: IDEON^{TM}: an extensible ontology for designing, integrating, and managing collaborative distributed enterprises. Syst. Eng. **4**(1), 35–48 (2001)
45. Mazouni, M.H., Aubry, J.F.: A PHA based on a systemic and generic ontology. In: 2007 IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 1–6. IEEE (2007)
46. Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I.: Internet of Things: vision, applications and research challenges. Ad Hoc Netw. **10**(7), 1497–1516 (2012)

47. Nagowah, S.D., Sta, H.B., Gobin-Rahimbux, B.A.: An overview of semantic interoperability ontologies and frameworks for IoT. In: 2018 Sixth International Conference on Enterprise Systems (ES), pp. 82–89. IEEE (2018)
48. Palmer, C., Urwin, E.N., Niknejad, A., Petrovic, D., Popplewell, K., Young, R.I.: An ontology supported risk assessment approach for the intelligent configuration of supply networks. J. Intell. Manuf. **29**, 1005–1030 (2018)
49. Pattar, S., et al.: Ontology based service discovery for intelligent transport systems using Internet of Things. In: 2018 Fourteenth International Conference on Information Processing (ICINPRO), pp. 1–3 (2018)
50. Ponomarov, S.Y., Holcomb, M.C.: Understanding the concept of supply chain resilience. Int. J. Logist. Manage. **20**, 124–143 (2009)
51. Protalinsky, O., Khanova, A., Shcherbatov, I.: Simulation of power assets management process. In: Dolinina, O., Brovko, A., Pechenkin, V., Lvov, A., Zhmud, V., Kreinovich, V. (eds.) ICIT 2019. SSDC, vol. 199, pp. 488–501. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12072-6_40
52. Ramezani, J., Camarinha-Matos, L.M.: Approaches for resilience and antifragility in collaborative business ecosystems. Technol. Forecast. Soc. Chang. **151**, 119846 (2020)
53. Ramzy, N., Auer, S., Ehm, H., Chamanara, J.: MARE: semantic supply chain disruption management and resilience evaluation framework (2022)
54. Rao, J., Gao, S., Miller, M., Morales, A.: Measuring network resilience via geospatial knowledge graph: a case study of the US multi-commodity flow network. In: Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Knowledge Graphs, pp. 17–25 (2022)
55. Russomanno, D.J., Kothari, C.R., Thomas, O.A.: Building a sensor ontology: a practical approach leveraging ISO and OGC models. In: IC-AI, pp. 637–643 (2005)
56. Scheuermann, A., Hoxha, J.: Ontologies for intelligent provision of logistics services, pp. 106–111 (2012)
57. Scheuermann, A., Leukel, J.: Supply chain management ontology from an ontology engineering perspective. Comput. Ind. **65**(6), 913–923 (2014)
58. Seydoux, N., Drira, K., Hernandez, N., Monteil, T.: IoT-O, a core-domain IoT ontology to represent connected devices networks. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) EKAW 2016. LNCS (LNAI), vol. 10024, pp. 561–576. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49004-5_36
59. Singh, S., Ghosh, S., Jayaram, J., Tiwari, M.K.: Enhancing supply chain resilience using ontology-based decision support system. Int. J. Comput. Integr. Manuf. **32**(7), 642–657 (2019)
60. Sinha, D., Roy Chowdhury, S.: A framework for ensuring zero defects and sustainable operations in major Indian ports. Int. J. Qual. Reliab. Manage. **39**(8), 1896–1936 (2022)
61. Suherman, A.G., Simatupang, T.M.: The network business model of cloud computing for end-to-end supply chain visibility. Int. J. Value Chain Manage. **8**(1), 22–39 (2017)
62. Suri, K., Gaaloul, W., Cuccuru, A., Gerard, S.: Semantic framework for internet of things-aware business process development. In: 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 214–219. IEEE (2017)
63. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The enterprise ontology. Knowl. Eng. Rev. **13**(1), 31–89 (1998)
64. Xu, L.D., He, W., Li, S.: Internet of things in industries: a survey. IEEE Trans. Industr. Inf. **10**(4), 2233–2243 (2014)

65. Ye, Y., Yang, D., Jiang, Z., Tong, L.: An ontology-based architecture for implementing semantic integration of supply chain management. Int. J. Comput. Integr. Manuf. **21**(1), 1–18 (2008)
66. Yeboah-Ofori, A., Mouratidis, H., Ismai, U., Islam, S., Papastergiou, S.: Cyber supply chain threat analysis and prediction using machine learning and ontology. In: Maglogiannis, I., Macintyre, J., Iliadis, L. (eds.) AIAI 2021. IAICT, vol. 627, pp. 518–530. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79150-6_41
67. Yodo, N., Wang, P.: Engineering resilience quantification and system design implications: a literature survey. J. Mech. Des. **138**(11), 111408 (2016)
68. Zdravković, M., Panetto, H., Trajanović, M., Aubry, A.: An approach for formalising the supply chain operations. Enterp. Inf. Syst. **5**(4), 401–421 (2011)
69. Zekhnini, K., Cherrafi, A., Bouhaddou, I., Benabdellah, A.C.: Suppliers selection ontology for viable digital supply chain performance. In: Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., Romero, D. (eds.) APMS 2021. IAICT, vol. 633, pp. 622–631. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85910-7_66
70. Zhou, F., He, Y., Ma, P., Mahto, R.V.: Knowledge management practice of medical cloud logistics industry: transportation resource semantic discovery based on ontology modelling. J. Intellect. Cap. **22**(2), 360–383 (2021)

A Model-Driven Approach to SAP S/4HANA Development

Jonathan Neugebauer[1(✉)], Jonas Hochstrat[1], Konrad Schneid[2], Daniel Sigge[2], and Herbert Kuchen[1]

[1] Department of Information Systems, University of Münster, Münster, Germany
{jonathan.neugebauer,j_hoch05, kuchen}@uni-muenster.de

[2] Best Practice Consulting AG, Am Mittelhafen 16, 48155 Münster, Germany
{konrad.schneid,daniel.sigge}@bpc.ag

Abstract. While Enterprise Resource Planning systems such as SAP S/4HANA play a key role for many companies, they rarely come alone but are connected to other applications via interfaces. Usually, interface development is done for each project individually. However, there can still be many common requirements shared by multiple projects causing repetitive coding and leading to a maintenance nightmare. In this work, we introduce a novel approach to SAP S/4HANA development driven by models from which running code can be generated automatically. Thus, repetitive coding is avoided and development effort reduced. To this end, we discuss different methods of importing externally generated code into SAP S/4HANA. This is contrary to the development style pursued traditionally and required an analysis of how different development objects must be represented to be importable. As a case study, we apply our approach to interface development. However, beyond this use case, we hope to see applications of our approach in various other areas in the future.

Keywords: Model-driven Software Development · Enterprise Resource Planning · SAP S/4HANA · Data Integration

1 Introduction

By implementing support for various business processes and managing related data within a single system, Enterprise Resource Planning (ERP) systems are crucial for many companies. Nonetheless, ERP systems seldom come alone but coexist with other services within a larger application landscape. Consequently, integration with external applications is often necessary involving both data imports and exports (e.g., business partners or financial documents).

In SAP S/4HANA[1], a widely used ERP solution, interfaces can be implemented using the ABAP programming language [11] to address the requirement of connecting external applications [17, pp. 401–417]. However, such interfaces

[1] https://www.sap.com/products/erp.html.

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 97–113, 2024.
https://doi.org/10.1007/978-3-031-46587-1_6

occur in a plethora of variations including different source types (e.g., file-based or API-based), data formats (e.g., CSV or XML), and data schemata (e.g., standardized such as SAF-T [8] or application-specific). Therefore, interfaces are often developed individually as part of the customizing process which is time-consuming and requires technical as well as business knowledge. While interface implementations typically cannot be re-used between projects without changes, there can still be many common requirements. This leads to re-occurring patterns which, in turn, may cause code duplications. Redundant code residing in various implementations, on the other hand, is challenging to maintain.

A well-proven method to simplify implementing similar artifacts for a specific domain is Model-Driven Software Development (MDSD) [19]. When following a model-driven approach, schematic repetitive code is not written manually but generated automatically based on a model specified in a Domain-Specific Language (DSL). In this work, we apply MDSD to interface development for SAP S/4HANA. However, beyond this specific use case, we pose a broader research question: How can SAP S/4HANA development be pursued based on a model-driven approach? We contribute to answering this question in multiple ways:

1. discussion on how *externally generated code* can be imported,
2. analysis of how different *development objects* can be represented as files,
3. introduction of a *domain model* for specifying different development objects,
4. design of a *model-driven development process* involving multi-stage code generation including model-to-model as well as model-to-text transformations,
5. application of this process within a *case study* on interface development.

The remainder of this paper is structured as follows. In Sect. 2, we outline how development is done traditionally and mention resulting challenges to our work. Then, Sect. 3 presents our model-driven approach which is applied within a case study in Sect. 4. In Sect. 5, we discuss our approach. Related work is mentioned in Sect. 6. Lastly, in Sect. 7, we conclude and point out future work.

2 Traditional Approach

Adding custom business logic to SAP's application server requires writing ABAP code. However, programming in ABAP is different from what might be known from other programming languages. Development is done exclusively against a running SAP system. Furthermore, ABAP code is not stored in separate files on the filesystem but resides in the SAP database. Next to the plain source code ("development object"), a transformed byte code representation is saved ("runtime object"). While the former is used for code editing, the latter is used for execution in the ABAP runtime environment (similar to Java byte code executed in the Java Virtual Machine) [9,13].

There are two sets of development tools from which an ABAP developer can choose. *ABAP Workbench* [12] provides an editing environment that is directly embedded into the SAP system. It can be accessed from within the SAP GUI which is also employed by the end users for their daily work. As a modern

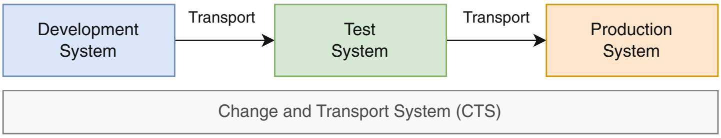


Fig. 1. Propagation of Changes Between Different Environments via Transports.

alternative, the *ABAP Development Tools* [16] integrate into the Eclipse IDE[2] for remote editing of development objects. While ABAP development generally involves writing source code, there are still various tasks that can only be performed in dialog windows (e.g., defining customizations that are stored as table entries without a natural source code representation).

Usually, multiple environments are used for different phases of the development process as illustrated in Fig. 1. In the *development system*, all ABAP programming takes place before changes are propagated to the *test system* for quality assurance. Finally, changes are applied to the *production system*. A set of changes can be released and copied to the next stage via *transports*. This process is fully managed by the *Change and Transport System (CTS)* tool. Hence, exporting and importing changes manually is not required [14].

The traditional approach makes applying model-driven methodologies challenging as SAP fully handles the code which, in turn, means that ABAP code is usually never represented as code in a file. Given the nature of a code generator which follows the idea of generating source code in the form of text-based files, the absence of files in ABAP development is a significant problem to address. As a consequence, the following issues have to be resolved to successfully introduce a model-driven development approach:

- representing ABAP development objects as text (including elements not specified with ABAP code but using dialog windows),
- generating files containing such objects,
- importing these externally generated files into SAP S/4HANA.

3 Model-Driven Approach

In this section, we introduce our model-driven development approach. But before outlining the novel development process in detail in Subsect. 3.4, Subsects. 3.1 to 3.3 revisit the issues mentioned in the previous section. First, different methods of importing development objects as files are discussed. Then, we present an analysis of how development objects can be represented as files based on the import method we chose. Lastly, we introduce a domain model for specifying development objects.

[2] https://eclipseide.org.

3.1 Importing Development Objects as Files

We considered multiple methods to import externally generated development objects as files. The following paragraphs provide a discussion.

Uploading a Transport Request File. As mentioned previously, changes can be propagated along system environments via transports (cf. Fig. 1). To initiate a transport in the CTS tool a *transport request* can be configured bundling a set of changes [14]. There is an unofficial way of downloading a transport request file[3] and uploading it to another system[4]. Transport request files are archives containing all information on the objects to be transported. However, besides this method not being an officially supported feature, a brief analysis of the file structure has shown an uncommon encoding rendering the file unreadable. Thus, trying to implement a code generator for transport request files is considered impractical.

Pasting Into ABAP Editor. A naive approach is to generate ABAP source code externally to copy-and-paste it into one of the mentioned code editors. This method would work for, e.g., classes that can be edited in a source code-based mode [12]. However, this method cannot be applied to all development objects. As previously mentioned, certain aspects can only be defined using dialog windows. This limitation renders this approach unusable for importing generated code. Moreover, having to manually copy-and-paste updated code after every generator run is very tedious.

Batch Input Sessions. As an addition to the previous method, batch input sessions could be used to imitate user input into dialog windows [15]. This way, the remaining development objects that cannot be created by pasting source code into the ABAP Editor could be defined programmatically. However, we consider this method as being too complex and lacking the robustness for everyday development work as it is rather designed for one-time system migrations.

Extending the ADT Plugin. Eclipse's functionalities can be extended by implementing plugins as with ADT. We considered developing an additional plugin extending ADT's functionalities. First, this would require connecting a code generator to the ABAP editor provided by ADT. However, as discussed previously, finding a method to create non-source-code-based development objects programmatically would also be necessary. This would require an inspection of the ADT implementation. We discarded this idea for being too complex and not well-documented.

Deploying via Git Repositories. A fundamentally different method is based on the version control system Git[5]. As mentioned in the previous section, CTS is handling version control in traditional ABAP development. However, Git can be integrated for this purpose as well. As a Git repository can be

[3] https://blogs.sap.com/2013/08/24/how-to-download-upload-transport-request-from-to-a-server/.

[4] https://blogs.sap.com/2019/08/19/upload-a-sap-transport-request-made-easy/.

[5] https://git-scm.com.

accessed from outside SAP, a code generator could place the generated files into the repository.

There are two tools allowing to deploy code changes via Git repositories. We elaborate on both.

abapGit[6] is an open-source tool enabling Git-based development workflows within SAP S/4HANA. It allows connecting an ABAP package to a Git repository for synchronization. Consequently, changes can be pulled from the repository into the system or pushed from the system to the repository. abapGit supports various ABAP object types[7] which are mapped to files for synchronization with the repository. Besides source code (stored in `.abap` files) also metadata (stored in `.xml` or `.json` files) is saved in the repository. By consulting the corresponding reference documentation[8] and inspecting the file structure a code generator for ABAP development objects could be implemented. However, abapGit's features are only available after importing third-party code which introduces security concerns. A user should not have to install potentially untrusted third-party software to utilize the model-driven development approach. Thus, using abapGit is not considered further.

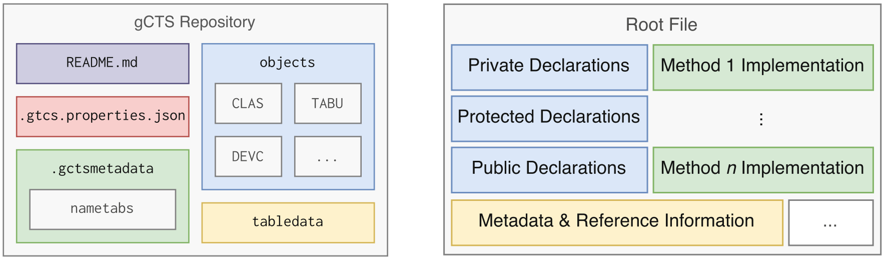
Git-enabled Change and Transport System (gCTS) [14] is a built-in tool available in recent versions of SAP S/4HANA. Hence, no additional installation is required. gCTS works similarly to abapGit as it allows pushing and pulling development objects to and from Git repositories. It is integrated into the CTS tool mentioned previously. A single repository can be linked to multiple system environments. For instance, changes can be pushed from the development system to the repository and pulled to the test and production systems (where the repository can be configured in read-only mode). Besides development objects, gCTS also supports the handling of database records which helps deploy changes in customizing tables. Similar to abapGit, the file structure created by gCTS mainly consists of `.json` and `.abap` files. These features of gCTS and the fact that it is a built-in tool make it the most promising method of importing development objects as files. However, before a code generator targeting gCTS repositories can be implemented, the repository file structure has to be analyzed.

3.2 Representing Development Objects as Files

This subsection deals with the file structure that gCTS maintains in the Git repository. SAP is working on standardizing how development objects are represented as files[9]. However, as this is ongoing work, gCTS does not yet follow the proposed format. Unfortunately, there is no documentation on how gCTS implements the translation to files. Hence, a manual analysis of the file structure is required. For this purpose, we created a new ABAP package linked to a Git

6 https://abapgit.org/.
7 https://docs.abapgit.org/user-guide/reference/supported.html.
8 https://docs.abapgit.org/user-guide/reference/folders-filenames.html.
9 https://github.com/SAP/abap-file-formats.

repository and created various development objects relevant to this work. After adding a particular object we examined the file changes that gCTS pushed to the repository. This way, we were able to analyze which files are needed to specify the respective object.



(a) Repository Structure. (b) Class Structure.

Fig. 2. Structure of a gCTS Repository (simplified).

Subsequently, we elaborate on our findings. An exemplary gCTS repository can be found in SAP's samples collection on GitHub[10]. As illustrated in Fig. 2a, a gCTS repository has the following contents (next to a README file).

Properties File. This file contains configuration properties concerning the repository [14]. For instance, here the user can enable that changes in customizing tables are tracked.

Metadata Folder. In this folder, meta information is stored [14]. Relevant to this work is the `nametabs` subfolder containing a `.json` file for every SAP table that is needed to store or handle the objects in the repository.

Objects Folder. In this folder, development objects are stored. It contains subfolders for each object type. For instance, ABAP Packages are specified by `.json` files within the `DEVC` folder. Definitions of ABAP classes, on the other hand, reside within the `CLAS` folder and have a more complicated structure. As can be seen in Fig. 2b, the source code for classes is split into multiple `.abap` files which are referenced in a root file. The file structure resembles the source code structure of classes consisting of a *declaration part* (containing declarations of attributes, methods, and events) and an *implementation part* (where methods are implemented) [11,13]. While the declaration part is split into three files based on visibility (i.e., private, protected, public), the implementation part is split into one file for each method. There are also some additional files for other elements not discussed further (e.g., files for helper classes, macro definitions, or unit test classes). Each class requires various entries in internal SAP tables. This includes metadata about the class (e.g., author or description) as well as reference information (e.g., class

[10] https://github.com/SAP-samples/s4hana-gcts.

attributes or method parameters). Such entries are stored in a central `.json` file accompanying the class.

Table Data Folder. When activated in the properties file, customizing data is stored in the table data folder as `.json` file specifying the name of the table as well as the values for each of its attributes. Additionally, for each table, a `.json` file is created in the `TABU` subfolder within the objects folder containing all index columns for the respective table entry.

Our analysis showed that development objects in the presented gCTS format follow a well-defined structure that can be created with the aid of a code generator given the necessary information on an object's contents. In the next subsection, we discuss which information is required as input to a code generator.

3.3 Domain Model

The domain model is derived from the presented structure of gCTS repositories and is designed to serve as input for the code generator. A corresponding UML class diagram is depicted in Fig. 3. A prototype implementation is available on GitHub[11]. It will be further discussed in the following subsection and is the basis for the use case that we present in the upcoming section. For better readability, some element names have been simplified (e.g., as the domain model was implemented in Java, names colliding with Java keywords are adapted in the implementation). In the following, we elaborate on the domain model's elements. Attributes required to generate the respective files in gCTS format are indicated. Some entities are specific to a particular client within the SAP system and, thus, contain a corresponding attribute. The implemented domain model contains further classes allowing to generate objects for ABAP development, however, we focus on the elements most relevant to this work.

The central element of the model is the *repository*. To this entity, all other elements are added. A repository may contain multiple packages, classes, and customizations. Different metadata is required to specify a *package* (e.g., name and author). Additionally, a package may have a parent package. For *classes*, similar metadata is required. Furthermore, each class is assigned to a package and optionally has a superclass as a parent. In addition, classes may contain *methods*, *attributes* and *data types*. These three entities are specified with a name, visibility, and a body containing the required source code as text. Methods, in turn, have an additional attribute to specify whether a method of the superclass is redefined. Moreover, methods may have *parameters* defined by their name, parameter type (i.e., "importing" for inputs, "exporting" for outputs, and "raising" for exceptions), and data type. Additionally, it can be specified whether the parameter should be passed by reference or by value. Lastly, to define *customizations* the respective customizing table, a list of key attributes as well as a map of column names and corresponding values must be given.

[11] https://github.com/wwu-pi/mabap.

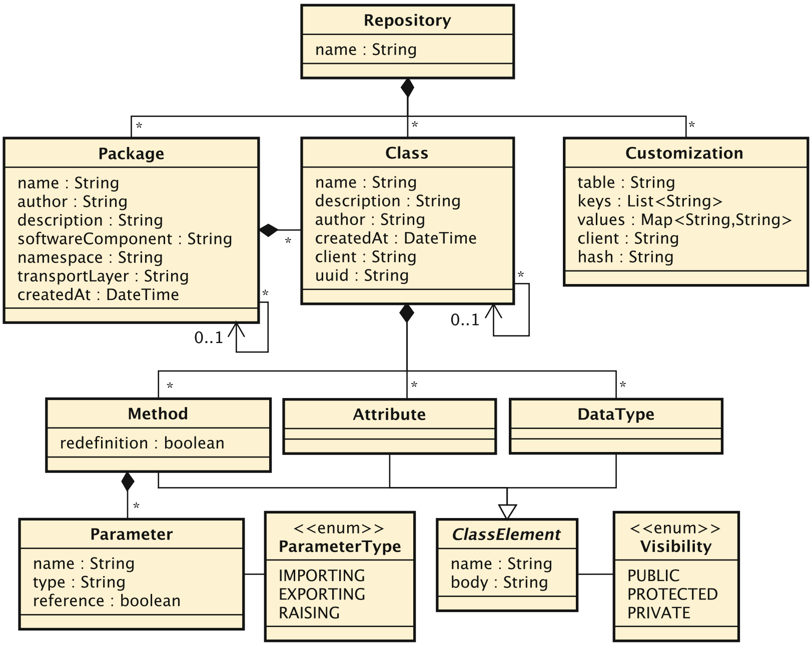


Fig. 3. Domain Model Serving as the Input for the Code Generator (simplified).

There are two specific attributes in the domain model which require a more detailed discussion. First, the attribute *UUID* is required for table entries accompanying a class definition. Normally, this attribute is generated automatically by SAP's development tools. However, it can also be generated manually using the ABAP class `CL_SYSTEM_UUID`. Second, each JSON file in the *TABU* folder required for defining entries in customizing tables (cf. Fig. 3) contains a hash value that could not be generated but reused from other files without impacting the import process.

3.4 Development Process

In this section, we present the envisioned model-driven development process. An illustration of the process can be found in Fig. 4. Here, we refer to the domain model discussed in the previous subsection with the term *repository model* to distinguish it from another domain model called *application model*. While the former is suited to model the domain of gCTS repositories, the latter is specific to the use case in which the development takes place. For instance, interfaces for SAP S/4HANA could be defined within an application model. We revisit the idea of specifying interfaces in the application model in the upcoming section when we present a case study based on our proposed development process.

The process starts with the user specifying the application model with the aid of a *DSL*. The user is provided with an *editing environment* providing sup-

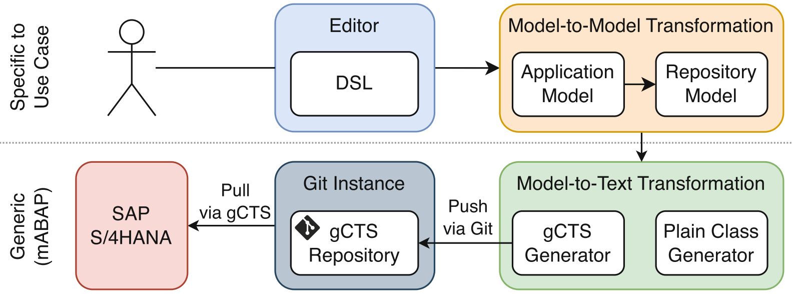


Fig. 4. Model-Driven Development Process.

port for the DSL (e.g., syntax highlighting, validation, and code completion). Depending on the use case, different DSL types may be suitable (e.g., textual or graphical). Then, the DSL code is parsed into the application model which is transformed to the repository model. This *model-to-model transformation* is implemented by a translator. The repository model, in turn, is used as the input for a *model-to-text transformation* implemented by a generator. We implemented two generators. First, a *gCTS generator* creating the necessary files corresponding to the repository model in the presented gCTS format. Second, a *plain class generator* outputting the ABAP source code for classes. This plain variant is comparable to the format displayed when developing in ADT or using the source-code-based view in the ABAP workbench. However, the source code is not split into multiple files as it is done by the gCTS generator. This facilitates debugging the generated ABAP code. Lastly, the generated gCTS repository is *imported into SAP S/4HANA* by pushing it to a Git instance and pulling the objects via gCTS. Alternatively, the output of the plain class generator could be used for SAP systems without support for gCTS by copying and pasting the source code into the traditional development tools.

In Fig. 4, the top and bottom parts are separated. While the top part is specific to the use case and has to be implemented to suit each application, the lower part is generic and fully covered by our prototype implementation. We named the prototype `mABAP` which stands for **m**odel-driven **ABAP**. The code generators were implemented with the language engineering framework Xtext[12] based on Xtend[13] templates.

One advantage of this modularized code generation approach involving two transformations is that alternative generators can be added at any time based on the repository model without changing the other parts of the process. For instance, a generator for abapGit could be added this way. Additionally, `mABAP` can be reused for various applications one of which will be discussed in the next section.

[12] https://www.eclipse.org/Xtext/.
[13] https://www.eclipse.org/xtend/.

4 Case Study

In this section, we present a case study carried out with the company best practice consulting AG. We apply the model-driven approach introduced in the previous section. First, we outline the use case (Subsect. 4.1). Then, Subsect. 4.2 introduces a corresponding domain model (representing the application model of Fig. 4). Lastly, we describe the development tools implemented within this case study in Subsect. 4.3.

4.1 Use Case

For the case study, we revisit the interface development use case introduced in the beginning. Here, we focus on file-based interfaces. Implementing an interface in SAP requires four steps. First, a *file upload* must be realized. Second, the uploaded file must be *parsed* into an intermediary representation suitable for further processing (usually table entries a program can iterate over). Third, the data has to be *mapped* to reach a format allowing to perform the last step: *object creation*. Mapping the data is often the most complex step, as there can be various issues that need to be addressed (e.g., format transformations, inconsistencies or missing data). Additionally, there can be differences in the structure and dimensions between the source and target of the import requiring further transformation steps.

At best practice consulting AG, interface development for SAP is facilitated by an internal framework supporting the implementation of the four mentioned steps. The import process is realized within a *converter class*. After uploading, parsing, and mapping the input data, the converter creates so-called *interface objects*. Interface objects represent a standardized intermediate format that can be further processed by the framework. Every new interface requires a new converter suitable for the structure of the input data and the target object. However, similar patterns and repetitive code can be found in the implementation logic when comparing different converter classes. This renders converters a suitable target for code generation.

4.2 Domain Model

The domain model for specifying converters was realized with a textual DSL. A model instance consists of specifications regarding three main areas. In the following, we elaborate on each of these areas. A corresponding UML class diagram is depicted in Fig. 5.

Interface Structure. The main part of the model is the specification of the interface structure. The structure can consist of multiple *objects* which may also be nested (e.g., invoice documents containing multiple positions). Additionally, there can be *variations* of objects requiring different processing. For instance, an interface for invoice documents could require two variations:

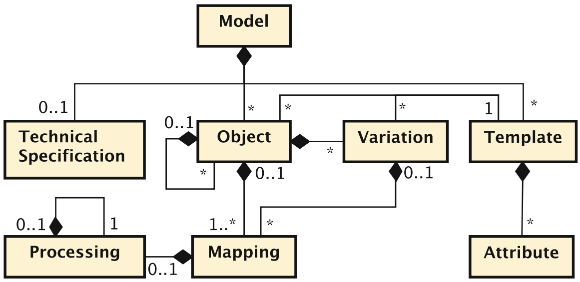


Fig. 5. Domain Model for Specifying Converters (simplified).

one for incoming and one for outgoing invoices. Based on a specified condition, the processing is then adapted to the recognized invoice type. *Mappings* define the assignment between input attributes and target attributes. They can optionally involve *processings* allowing predefined data transformations (e.g., replacing text or selecting a substring). Moreover, custom logic can be added by injecting ABAP code. Processings can also be nested when more complex transformation chains are needed.

Technical Specification. To enable code generation, a *technical specification* regarding the interface and the resulting repository has to be provided (e.g., input file type, class names, or other ABAP-related details). However, following the idea of separating specification from implementation, these details are optional. Thus, non-technical users can define the object structure and mappings and developers can add implementational details later.

Templates. As outlined previously, converters create interface objects based on the input data. In the process of defining the interface structure, a user would benefit from suggestions and validations regarding the attributes of the targeted interface object. However, the DSL is fully detached from regular ABAP development and, thus, is not aware of the defined interface objects. As a solution, *templates* together with their corresponding *attributes* can be created and assigned to an object or a variation. Such templates can be defined centrally and reused for different model instances.

In Listing 1, a simplified exemplary converter specification for importing invoices from CSV files can be seen. The *interface structure* is defined in lines 4 ff. For each field, the column from the input file is specified as well as a name for referencing within the specification. The mapping to a corresponding target field is defined after the keyword `to`. For instance, based on line 8, a `type` field imported from column 2 is mapped to the target field `dtype`. Additionally, the `type` field is used to distinguish between incoming and outgoing invoices in the `Variations` block (cf. lines 12 ff). Here, based on the type either the customer or vendor number is imported. Moreover, the example contains a nested object specification (cf. lines 21 ff) to account for importing invoice positions. Examples of processings can be seen in lines 7 and 23. In this case, no *templates* have been

```
1 Interface invoice_interface {
2   DataSource: { File type CSV as Upload }
3
4   ImportStructure: { invoice }
5
6   Object invoice {
7     doc_no from UpperCase(col(1)) to dno
8     type from col(2) to dtype
9     vendor_no from col(3)
10    customer_no from col(4)
11
12    Variations {
13      when(type is "incoming"): Variation inc_invoice {
14        vendor_no to vno
15      }
16      when(type is "outgoing"): Variation out_invoice {
17        customer_no to cno
18      }
19    }
20
21    Object position {
22      item_no from col(4) to ino
23      amount from Replace(col(5), ",", ".") amnt
24    }
25  }
26 }
```

Listing 1. Exemplary Converter Specification (simplified).

used and most *technical specifications* have been omitted except for line 2 defining that for the input, CSV files are provided by uploading them to SAP S/4HANA.

4.3 Development Tools

The DSL corresponding to the presented domain model has been implemented with the aid of the Xtext framework. Additionally, an editing environment for the DSL is provided based on an Eclipse plugin. Thus, the Eclipse editor known to ABAP developers from traditional development with ADT can also be used for the model-driven approach. The plugin supplies features such as syntax highlighting, code completion, and validation. Lastly, we implemented the model-to-model transformation from the application model to the repository model which concludes our case study since the previously presented prototype `mABAP` implements the remaining steps (cf. Fig. 4).

5 Discussion

In this section, we discuss the results of our work based on the case study presented in the previous section. First, we evaluate the effectiveness of the model-driven development approach with a lines of code (LOC) comparison (Subsect. 5.1). Then, Subsect. 5.2 provides insights gained from interviewing employees working as SAP consultants and in ABAP development. Lastly, we assess the usability of the model-driven approach in Subsect. 5.3.

5.1 Lines of Code Comparison

We conducted a LOC comparison to examine to which extent the model-driven approach used in the case study can reduce the development effort required in interface development. More specifically, we compare the *physical LOC* that have been written in the DSL specifications manually with the physical LOC generated automatically by the `mABAP` code generator. Physical LOC are the lines not containing blanks or comments [6]. As explained previously, except for classes development objects are created by using dialog windows in traditional development making a LOC comparison impossible. But as the development effort in creating interfaces mainly focuses on the implementation of the converter class, we calculate the LOC metric based on the generated class files.

We examined two exemplary implementations. The corresponding DSL models can also be found in the mentioned GitHub repository. First, we developed a simple interface for *fund reservations* by using the DSL. It processes the reservation of funds from budgets in anticipation of future payments. The structure consists of one main object that may have multiple positions. Consequently, two object types and a total of 39 mappings were specified in the model. The second example is an interface for importing *financial documents*. Here, three object types, eight variants, and a total of 65 mappings were needed. In contrast to the first example, this implementation also required specifying custom ABAP code in the model for more complex processing.

In Table 1, the results of the LOC comparison are summarized. As can be seen, by following the model-driven development approach the required LOC can be reduced substantially (reduction by 86.7% and 82.5%, respectively). Here, we did not count the LOC for templates as these can be reused for every new interface and are technically not required for code generation (only for assistance in the editor). However, the reduction is still considerable even when also counting the LOC for templates as they accounted for only 44 and 166 LOC, respectively. This indication is affirmed by qualitative feedback from the interviews that we present in the next section.

5.2 Semi-structured Interviews

We interviewed two employees at best practice consulting AG working as SAP consultants including 60–75% development activities. The range of experience in SAP interface development ranges from two to ten years. Both persons work on

Table 1. LOC Comparison for two Exemplary Implementations.

Language	Funds Reservation		Financial Documents	
	Traditional	Model-Driven	Traditional	Model-Driven
ABAP	678	0	1,079	34
DSL	0	90	0	154
Sum	678	90	1,079	188

interface-related topics at least weekly with one participant even stating doing that daily. The interviews were conducted as semi-structured interviews [1]. Similar to a real-world project, the participants were given requirements for implementing a new interface. The test scenario consisted of an interface for financial documents requiring the usage of all features implemented in the prototype. We also provided documentation on using the prototype. After studying the materials the participants were asked how they would implement the required interface using traditional means. After that, the attendees implemented the interface using the prototype while explaining to the interviewer which steps they undertake as well as their reasoning behind it (similar to a think-aloud test [18]). After completing the implementation, the participants were asked to describe their experience. In the following, we summarize the main insights received from the interviews structured in different categories.

Validity. Both participants stated that the scenario is comparable to a real-world project while one participant added that the complexity of data mapping is usually higher with more complicated requirements. This criticism is expected as the scenario had to be designed in such a way that participants have a realistic chance to solve it during the interview.

Generalizability. The participants stated that the prototype can be used for most interfaces that process financial documents as these share a similar processing logic. Above that, the prototype would also be usable for other interface types when their structure is similar to financial documents which—according to one employee—is the case for many of the interfaces developed over the last years. Thus, the participants assess good generalizability.

Effectiveness. Both attendees stated that the prototype reduces the writing effort. Given some training, the time required to develop a new interface could likely be reduced from one working day to two hours (reduction by 75%). Additionally, one participant mentioned that using the prototype has a positive impact on code quality and a higher level of standardization in the development process.

Understandability. Initially, the mentioned idea of separating specification from implementation (cf. Subsection 4.2) was intended to make the model understandable for non-technical users. However, the participants perceived the understandability for that user group to be limited. Technical knowledge is still required even for the definition of mappings.

Table 2. Ratings for the SUS Statements.

	A	B
1. I think that I would like to use the system frequently	4	**3**
2. I found the system unnecessarily complex	2	1
3. I thought the system was easy to use	4	**3**
4. I think that I would need the support of a technical person to be able to use the system	2	**5**
5. I found the various functions in the system were well integrated	5	4
6. I thought there was too much inconsistency in the system	2	1
7. I would imagine that most people would learn to use the system very quickly	4	4
8. I found the system very cumbersome to use	2	2
9. I felt very confident using the system	4	4
10. I needed to learn a lot of things before I could get going with the system	**3**	**4**
SUS Score	75	62.5

Usability. Both participants stated that the general interface structure expressed in the domain model consisting of objects, variations, and mappings is an improvement to traditional development where this information must be defined repetitively. Also, the concept of templates and the assistance provided by the editor plugin were mentioned positively. However, the syntax of the DSL was perceived as not being intuitive in all parts. For instance, the usage of processings and variations required additional explanations by the interviewer. Moreover, as in traditional ABAP development the usage of Git is not necessary, the interview partners did not have much experience with the Git-based workflow and initially needed some time to get used to the import process. Additionally, debugging the generated code was difficult. As gCTS does not provide error messages, users have to check for the existence of the required development objects to verify a successful import. In case of a problem, they must consult the CTS logs or check the generated plain class for syntax errors.

5.3 Usability Test

To also measure the usability quantitatively, both interview partners filled out a System Usability Scale (SUS) questionnaire [3] after the interview. SUS is a standardized questionnaire for evaluating the usability of a system. It presents ten statements a participant must quantify on a 5-point scale ranging from strong disagreement to strong agreement.

The ratings of both participants can be seen in Table 2. Since the statements alternate between positive and negative sentences, high values on odd statements positively influence the score, while high values for even statements reduce it. The values with the most negative impact on the overall score are printed in bold. Notably, the most negative impact originates from statements 4 and 10 which concern the technical knowledge and the level of learning required to use the system. This finding aligns with the qualitative feedback gathered in the interviews. The resulting SUS scores can be interpreted based on the adjective rating scale proposed in [2]. While participant A assessed a “good” usability, participant B perceived the usability as being “ok”. The gathered results can be used as a foundation for future, more extensive evaluations.

6 Related Work

In [5], we propose a model-driven approach for customizing the ERP system Microsoft Dynamics 365 Business Central (BC). We present a DSL for specifying modules implementing new business processes based on which runnable code can be generated. However, as the DSL is modeling concepts specific to BC, it is not suited for SAP S/4HANA development.

Wolff and Bieler [20] suggest a model-based configuration interface for the ERP system iDempiere. They specify the configuration of users and roles within a model which is then converted to a data format suitable to be imported into iDempiere. However, their approach focuses on configuration and, thus, the development of new business logic is not in the scope of their work.

Dugerdil and Gaillard [4] present a model-driven solution for configuring the ERP system Adonix. Their approach is based on the Model-Driven Architecture (MDA) [7] which—analogously to this work—involves model-to-model and model-to-text transformations. However, similar to the previously mentioned work their focus is on configuration only.

Another application of the MDA can be found in [10] where the authors transform ERP process models to code that can be deployed to a process engine. As the approach focuses on generating code for an external system surrounding the ERP system, it is not suited for SAP S/4HANA development.

7 Conclusion and Outlook

This paper introduced a novel approach for SAP S/4HANA development. After outlining how development is done traditionally we presented a model-driven development approach based on the `mABAP` prototype. This involved defining a domain model for specifying required ABAP development objects as well as finding a way of representing such objects as files and importing them into SAP S/4HANA. We applied our approach within a case study on interface development. Then, we discussed this work based on a LOC comparison, semi-structured interviews, and a usability test. Lastly, we pointed out related work.

Future work on `mABAP` should primarily focus on improving error handling and debugging. For instance, tracing between generated code and DSL code would help identify error sources. Additionally, the generated ABAP code is validated late in the development process (only when the generated repository is imported). Future versions of `mABAP` could include linting of generated code to provide helpful feedback directly in the editor. Finally, the development of gCTS should be monitored closely and support for further development object types should be added.

References

1. Adams, W.C.: Conducting semi-structured interviews. In: Handbook of Practical Program Evaluation, pp. 492–505. Wiley, Hoboken (2015)
2. Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: adding an adjective rating scale. J. Usability Stud. **4**(3), 114–123 (2009)
3. Brooke, J.: SUS: a 'Quick and Dirty' usability scale. In: Jordan, P., Thomas, B., McClelland, I., Weerdmeester, B. (eds.) Usability Evaluation in Industry, pp. 189–194. Taylor & Francis, London (1996)
4. Dugerdil, P., Gaillard, G.: Model-driven ERP implementation. In: Proceedings of the 2nd International Workshop on Model-Driven Enterprise Information Systems - Volume 1: MDEIS, (ICEIS 2006), pp. 77–87. INSTICC, SciTePress (2006)
5. Neugebauer, J., Kuchen, H.: Model-driven customizing of ERP systems: a case study. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, SAC 2022, New York, NY, USA, pp. 1276–1279. Association for Computing Machinery (2022)
6. Nguyen, V., Deeds-Rubin, S., Tan, T., Boehm, B.: A SLOC counting standard. In: COCOMO II forum, vol. 2007, pp. 1–16. Citeseer (2007)
7. Object Management Group: MDA Guide Revision 2.0. Technical report, June 2014
8. OECD: Guidance for the Standard Audit File - Tax, Version 2.0. Technical report (2010)
9. Orosz, T.: Analysis of SAP development tools and methods. In: 2011 15th IEEE International Conference on Intelligent Engineering Systems, pp. 439–443 (2011)
10. Roser, S., Lautenbacher, F., Bauer, B.: MDSD light for ERP. In: Proceedings of the 2008 ACM symposium on Applied computing - SAC 2008, Fortaleza, Ceara, Brazil, p. 1042. ACM Press (2008)
11. SAP SE: ABAP - Keyword Documentation (2023). https://help.sap.com/doc/abapdocu_750_index_htm/7.50
12. SAP SE: ABAP Workbench Tools (2023). https://help.sap.com/docs/SAP_NETWEAVER_750/bd833c8355f34e96a6e83096b38bf192
13. SAP SE: Application Development on AS ABAP (2023). https://help.sap.com/docs/SAP_NETWEAVER_750/7bfe8cdcfbb040dcb6702dada8c3e2f0
14. SAP SE: Change and Transport System (2023). https://help.sap.com/docs/ABAP_PLATFORM_NEW/4a368c163b08418890a406d413933ba7
15. SAP SE: Managing Batch Input Sessions (2023). https://help.sap.com/docs/SAP_NETWEAVER_750/1e9c6cd6e04949458e31f0614eefd5f4
16. SAP SE: SAP - ABAP Development User Guide (2023). https://help.sap.com/docs/SAP_NETWEAVER_750/c238d694b825421f940829321ffa326a
17. Sarferaz, S.: Compendium on Enterprise Resource Planning: Market, Functional and Conceptual View based on SAP S/4HANA. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-93856-7
18. Someren, M., Barnard, Y., Sandberg, J.: The Think Aloud Method - A Practical Guide to Modelling Cognitive Processes, January 1994
19. Stahl, T., Völter, M.: Model-Driven Software Development: Technology, Engineering, Management. Wiley, Hoboken (2006)
20. Wolff, F., Bieler, K.: Evolutionary student research projects in domain specific modelling for an ERP-system with ADOxx. In: Bork, D., Grabis, J., Lantow, B. (eds.) Proceedings of the 2nd International Workshop on Practicing Open Enterprise Modelling within OMiLAB (PrOse). CEUR Workshop Proceedings, vol. 2238, pp. 13–24. CEUR-WS.org (2018)

Enterprise Analysis and Improvement with Process Mining

A Methodology for the Analysis of Robotic Systems via Process Mining

Flavio Corradini[1], Sara Pettinari[1](✉), Barbara Re[1], Lorenzo Rossi[1], and Francesco Tiezzi[2]

[1] School of Science and Technology, University of Camerino, Camerino, Italy
{flavio.corradini,sara.pettinari,barbara.re,lorenzo.rossi}@unicam.it
[2] Dipartimento di Statistica, Informatica, Applicazioni, University of Florence, Florence, Italy
francesco.tiezzi@unifi.it

Abstract. Robotic systems are widely adopted in various application scenarios. A very complex task for developers is the analysis of robotic systems' behavior, which is required to ensure trustworthy interaction with the surrounding environment. Available analysis techniques, like field tests, depend on human observations, while automated techniques, like formal analysis, suffer from the complexity of the systems. Recent works show the applicability of process mining for the analysis of event data generated by robots to increase the understanding of system behavior. However, robots produce data at such a low granularity that process mining cannot provide a meaningful description of the system's behavior. We tackle this problem by proposing a process mining-based methodology to prepare and analyze the data coming from the execution of a robotic system. The methodology supports the system developer in producing an event log compliant with process mining techniques and is used to analyze multiple perspectives of robots' behavior. We implemented the methodology in a tool supporting its phases. We use the tool on a robotic smart agriculture scenario to evaluate the feasibility and effectiveness of the methodology.

Keywords: Robotic Systems · Process Mining · Multi-Perspective Event Logs · Control-Flow Analysis · Spatial Analysis

1 Introduction

Nowadays, robotic systems are increasingly present in our daily lives since they reduce the need for human intervention in many domains, e.g., agriculture, industry, and healthcare. Nevertheless, a meaningful analysis of the implemented behavior of these systems is often a cumbersome task since the execution of a robot strictly depends on the capability to control several peripherals, i.e.,

This work was supported by the financial support of the PNRR MUR project ECS_00000041-VITALITY.

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 117–133, 2024.
https://doi.org/10.1007/978-3-031-46587-1_7

sensors and actuators, in real-time and to manage its interaction with the environment [6]. In this regard, various methodologies can help the developer spot drawbacks in the robotic system by analyzing its behavior, i.e., when and how an activity has been done, to gather information useful for monitoring and optimizing the system. The most used methodologies in the literature range from formal verification [16], which is often unfeasible due to the system complexity, to quantitative analysis [6], which requires humans in the constant observation system's execution.

More recently, to overcome these drawbacks, some works [19,21] propose the use of process mining for analyzing the behavior of robots through process models extracted from event logs generated during the execution of robotic systems. In these systems, process mining aims to address the complexity of understanding the behavior of the robots and their interactions [21] while checking the correctness of the whole system's execution [19]. Despite its potential, process mining has not been extensively investigated due to the complexity of transforming robotic event data into event logs. Indeed, robots collect continuous flows of data sent or received by their peripherals. Altogether these event data shape the robots' behavior from the point of view of the peripherals usage in the form of sequences of acting and sensing events with a very low-level of granularity and without correspondence with meaningful activities. In this regard, proper preparation and combination of data from multiple peripherals can identify a high-level robotic activity with a clear meaning (e.g., moving to a destination) suitable for applying process mining techniques [19].

Moreover, robots enable the possibility of recording context data, e.g., spatial information, resource usage, and exchange of messages; therefore, they generate multi-perspective event logs [14]. Among the others, the spatial perspective strongly impacts the control-flow, since robots may act in different parts of an unknown environment [5]. For instance, a robot programmed to find a target in an unknown environment could fail due to an obstacle or it can end its process with different outcomes. This relation between space and control-flow leads to the need for novel process mining techniques that go beyond process discovery [28]. However, a way to specify these perspectives of the process behavior has not been defined yet. Indeed, the spatial perspective plays a central role in robotic systems, while, to the best of our knowledge, process mining techniques do not consider such information.

To ease and foster the application of process mining for analyzing robotic systems, in this paper, we propose **a process mining-based methodology to prepare and analyze multi-perspective event logs extracted from executions of a robotic system**. In a nutshell, the methodology lets the robotic system developer tag relevant parts of the source code that correspond to the implementation of a robotic activity. This permits the selection of the desired level of abstraction, which depends on the granularity of the activities the developer identifies. While running the system, the developer's tags are triggered producing events with all the data about the system perspectives of interest, hence generating a log suitable for process mining. Then, process mining analysis techniques are applied to extract the behavior of the robot and provide significant insights regarding multiple perspectives. The methodology has been implemented in a tool we use to graphically show the results of the conducted analysis and evaluate the methodology. We assess the advantages of the proposed methodology in a robotic smart agriculture scenario, realistically reproduced in a simulation environment, that aims to analyze robotic control-flow and related spatial occupancy.

The rest of the paper is organized as follows. Section 2 introduces concepts of robotic systems and process mining and discusses their relation. Section 3 illustrates our methodology and the steps composing it. Section 4 discusses and evaluates the application of the methodology in a robotic system scenario. Section 5 presents the works related to the analysis of robotic systems, in particular in the application of process mining. Finally, Sect. 6 concludes the paper by discussing the results, possible extensions, and future works.

2 Bridging Robots and Process Mining

This section introduces the core concepts of our methodology: robotic systems, and process mining. The section also shows how event data coming from the execution of a robot can be adapted to process mining.

2.1 Robotic Systems

Robotic systems are emerging to automatize and speed up repetitive tasks such as assembling small parts in manufacturing processes and dealing with dangerous tasks for humans. Specifically, a robot is a complex system composed of several sensors and actuators that can fulfill a given mission by perceiving the surrounding environment and deciding how to act on it [4]. A robotic mission can be seen as the composition of multiple robotic activities that lead to the achievement of the goal, such as assembling a workpiece. In turn, a robotic activity, such as picking up an object, is managed by controllers capable of receiving data perceived by onboard sensors, making decisions, and sending commands to the actuators [15]. Each robot's activity involves the interaction of several low-level data shared between various sensors and actuators. Consequently, the development of a robotic system should deal with all the challenges brought by the interaction among all its components, such as distributed computation and coordination [11].

2.2 Process Mining

Process mining is a well-established discipline for analyzing systems behavior that consists of many techniques for discovering, monitoring, and improving the actual processes. Process mining relies on *event logs*, generated by the systems under investigation [1]. Event logs record data coming from systems' executions in an activity-centric manner, reporting at least two *attributes*, i.e., timestamp and activity name. Further attributes can be included in the events to provide context data that provides information about other perspectives. Events referring to the same system execution, i.e. *case*, are grouped into *traces* of events ordered by time. By analyzing such event logs, usually in the eXtensible Event Stream (XES) format, process mining reconstructs the process model describing the system's behavior, spots bottlenecks and unwanted deviations, and gets insights on the execution performances such as duration and costs [1].

Table 1. Event logs of the same robotic execution with different granularity

Time	Peripheral	Other Data
09:33:20.147	engine_1	...
09:33:20.147	camera	...
09:33:20.149	engine_2	...
09:33:21.239	gps	...
09:33:30.941	camera	...
09:33:32.410	gps	...
09:33:32.495	laser	...
09:33:32.499	laser	...
09:33:32:655	camera	...
09:33:33.003	engine_2	...
09:33:33.110	laser	...
09:33:33.118	laser	...
09:33:33.120	laser	...
09:33:33:240	camera	...
09:33:33.955	laser	...
...	...	...

(a) Peripheral-centric

Time	Activity	Lifecycle
09:33:20.147	Go To	start
09:33:20.147	Detect Target	start
09:33:33.240	Detect Target	complete
09:50:34.921	Go To	complete

(b) Activity-centric

2.3 Robotic Event Logs

The idea of using process mining to extract and analyze the behavior of robotic systems goes hand in hand with the existence of several works exploiting model-driven approaches for the specification of robots' missions [10,12,20]. Indeed, process models are intuitive enough to be understood by humans and formal enough to be deployed into real robots. Nevertheless, to use process mining in this context, we need to map the event data generated by a robotic system with the concept of the event log. During its execution, a robot produces event data in a *peripheral-centric* way, keeping track of data exchanged by sensors and actuators [9]. Specifically, a robotic peripheral-centric event is typically characterized by the *time* in which the event is produced, by the name of the *peripheral* that triggered it, and by other attributes storing additional data. For instance, a robotic event triggered by a laser sensor may store the distance data read as an attribute. These event data represent that the robot has accessed a peripheral functionality, whereas process mining techniques need event logs with an *activity-centric* structure, meaning that each event must represent a clear robotic activity, i.e., a high-level activity that should be executed to fulfill an objective in the process model, like a movement action to reach a destination.

To better clarify the difference between peripheral- and activity-centric event logs, we take as an example a wheeled robot equipped with engines, GPS, camera, and laser sensors, that has to move around to reach a destination (activity *Go To*) while looking for a target (activity *Detect Target*). Table 1a shows an excerpt of a peripherical-centric event log of one robot execution, whereas Table 1b reports the same event log abstracted to an activity-centric point of view. The peripheral-centric log describes the working flow of the robot peripherals, e.g., the first two rows of Table 1a indicate that the robot has used one engine and the camera. Instead, the activity-centric log provides a high-level view of the robotic execution, e.g., the first two rows of Table 1b show that the robot has concurrently started the Go To and Detect Target activities. To graphically visualize a correlation between the peripheral- and the activity-centric events and properly present the robotic granularity problem, we colored the events belonging to the Go To activity in purple, while the events correlated to the Detect Target activity are in yellow. In doing so, the obtained mixed occurrence of the colors in Table 1a shows that the peripheral events are produced in a continuous and unordered way, thus preventing the identification of a pattern for recognizing the related high-level activities. Moreover, the peripheral event log shows that both activities depend on the laser sensor, thus creating a problem if we want to identify whether an activity is finished.

Therefore, to properly apply process mining techniques, we need a methodology for producing an activity-centric event log from the execution of a robotic system. More in detail, a robotic event log should contain a collection of robotic executions, possibly started in different settings, in which a case contains sequences of high-level robotic activities recorded during a complete run.

3 The TALE Methodology

This section presents the methodology we designed to support the robotic developer in the automatic extraction of multi-perspective event logs from the execution of a robotic system and in their analysis through process mining. Figure 1 depicts the steps of the TALE (TAg-based muLti pErspective) methodology involving the event log preparation, and the event log analysis.

3.1 Preparation

The log preparation step aims to automatically extract an activity-centric event log suitable for applying process mining techniques. During the event log preparation, the source code of the robotic system is *tagged* by the developer in correspondence with the activities of interest. Then, the system is *executed* and, finally, the produced event data are *processed* to extract event logs. Below we analyze these three sub-steps in detail.

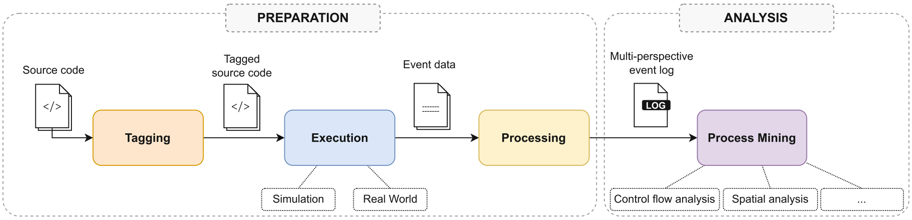


Fig. 1. The TALE methodology

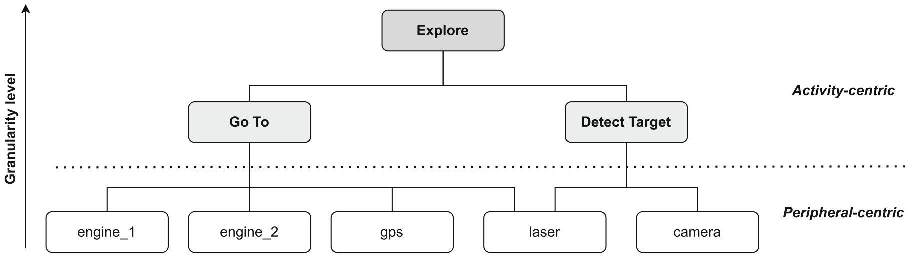


Fig. 2. Robotic activity composition

The first sub-step guides the developer in integrating tags in the source code, making it possible to determine where a relevant activity begins and finishes. Notably, the user performing this sub-step is typically the developer of the robotic system, who has complete knowledge of the code controlling the robots' behavior; therefore, the tagging integration is a task that would not require much effort from the developer.

The developer indicates points, i.e., tags, in the code that trigger events during the robot execution. Specifically, the developer specifies a list of attributes relevant to analyzing the robot mission with process mining. A tag has to report at least the name of the *activity*, and its execution status, i.e., the *lifecycle*, which either indicates the start or the completion status of the activity, i.e., *start* or *complete*. Moreover, since the execution of a robotic activity accesses peripheral data, the tag can be enriched with more attributes associated with other perspectives of interest. For instance, a tag could contain spatial coordinates, the battery level, or the message content exchanged with other devices. Notably, an essential characteristic is that the robotic activities must share the same and suitable level of granularity [29]. For example, in Fig. 2 we can see the composition of a high-level robotic activity named *Explore*, performed by a robot to inspect a specific area. This activity can be seen as the combination of two activities with a lower level of granularity, i.e., *Go To* and *Detect Target*, which in turn are realized through a combination of various peripheral-centric robotic data produced by a robot. Notably, the concept of tag directly related to the source code allows a developer to choose the desired level of granularity, hence passing from a peripheral-centric to an activity-centric point of view suitable for the analysis that will be applied to the event logs.

In the second sub-step, the code is executed multiple times in a simulation environment or a real-world scenario. Notably, a simulation environment allows for checking the system's correctness while avoiding deploying an incorrect behavioral model into physical devices. Additionally, the simulation eases and speeds up the massive execution of the system while allowing the testing with different environment settings. During the system execution, a listener catches the event data. Specifically, since the robot's perspectives, e.g., spatial position or battery level, can change during the execution of an activity, the listener records, in between the start and the completion of an activity, other events with the same activity name, a lifecycle status *inprogress* [18], and a series of values referring to the multi-perspectives of interest for the considered system.

Finally, in the processing sub-step, the recorded event data are processed to become an event log suitable for process mining analysis. TALE assigns a case with a unique identifier to the list of events recorded in each execution. Therefore, TALE prepares activity-centric event logs, where each recorded event represents a meaningful robotic activity and is enriched with all the perspectives the system developer chooses. The final result of this step is a multi-perspective event log compliant with the XES standard.

3.2 Analysis

Now we present the analysis step prescribed by TALE. Notably, since the methodology can be applied to produce multi-perspective robotic event logs, the choice of different perspectives affects the type of analysis that will be applied to them. Moreover, since the event logs are the typical input in process mining, several techniques can be applied, or developed, to improve the overall analysis [7]. At the time of being, we analyze the robotic event logs from two perspectives. The first one regards control-flow, whose analysis permits the discovery of the robot's behavior. The latter regards the space, whose analysis permits relating spatial information with the activities performed by the robot, to enhance the discovered behavior with additional perspectives.

Regarding the control-flow analysis, TALE adopts process discovery to provide a process model that describes the behavior of the robot, by observing the event log. The discovery algorithm, presented in [2], extracts from the log the events belonging to the robot and collects the directly-follows relations, i.e., binary relations indicating whether an activity occurs immediately after another. The relations between the activities of the robot are graphically represented as a Directly-Follows Graph (DFG), i.e., a directed graph whose nodes represent the process activities and the arcs indicate a directly-follows relation. In this form, the DFG summarizes the different executions of the robot seen in the event log. Each path from the start to the end node represents a sequence of activities the robot performs during the system execution. Moreover, the labels of the arcs report the number of times the connected activities have been executed one following the other.

Regarding the spatial analysis, TALE extracts from the log the x, y, and z coordinates of each event. These values represent the position in the environment occupied by the robot when performing its mission through points in a 3D space. In doing so, the activities in the discovered DFG are enhanced with information related to the spatial perspective. This enables the analysis of spatial conditions that have led to certain behaviors. Specifically, each activity refers to a group of points indicating the position of the robot when performing it. Moreover, the points can be filtered by the robot identifier, case, and activity name.

Summing up, altogether these techniques allow for spotting unknown relations between activities and space.

4 TALE in Practice

This section shows the feasibility and effectiveness of the methodology. For this purpose, we implemented a smart agriculture robotic system, made up of multiple robots, built upon the Robot Operating System (ROS) framework[1]. Then, we analyzed the event logs produced by the system with the process mining techniques prescribed by TALE. The tool implementing the methodology consists of two parts: one supporting the preparation step, and another supporting the analysis step. Notably, the source code for reproducing these steps and the results of the performed analysis are available online[2].

4.1 Smart Agriculture Scenario

The proposed scenario consists of one drone and two tractors that cooperate to identify and remove weed grass in farmland, thus increasing productivity. All the robots are equipped with a controller, a battery, and several sensors and actuators. At the system start-up, the drone takes off and starts the exploration of the field. During the flight, it can recognize weed grass areas, and when found, it sends the weed position to the tractors. This enacts the tractors, which send back to the drone their current position. The drone can hence elect the closest tractor and notify it. At this point, the selected tractor starts moving towards the field to reach the weed grass area and cut it. The drone execution can be interrupted if the drone battery runs out, or if the whole field has been explored and cleaned. At the same time, the tractors' execution can be interrupted by a low battery.

4.2 Robots Development Technologies

We built the smart agriculture scenario upon the ROS framework. Indeed, ROS is one of the most used frameworks for programming robots. It is designed as a framework of distributed nodes, i.e., processes able to perform computations

[1] https://docs.ros.org/en/foxy.
[2] https://pros.unicam.it/tale.

and designed to achieve a single purpose, e.g., activate motors or a laser sensor. Each node can communicate with the others by sending and receiving data following the publish-subscribe pattern, thus performing a *topic-based communication*. This enables a continuous data flow, typically used to share sensors' data. Notably, before deploying the ROS code into real robots, a good practice is to exploit a simulation environment. The reference simulator for ROS is Gazebo[3]. It permits rapidly designing and testing robotic applications in a safe, yet realistic, environment. Moreover, by exploiting the *ros2 bag*[4] library, it is possible to record the data generated during the robotic system execution. On this basis, we are going to discuss the application of the TALE methodology to the application scenario.

4.3 Preparation

To be compliant with the ROS architecture, we use ROS messages to contain the tag data. For this purpose, we defined a custom message that is suitable to store the required information. Specifically, it is composed of the *activity* field that identifies the action that the robot is performing, and the *lifecycle* that determines the status of the related activity. Then, we identified the activities performed by the robots choosing a level of abstraction suitable for the behavioral analysis. For the drone we selected *Take off*, *Explore*, *Weed found*, *Weed position*, *Tractor position*, *Closest tractor*, *Low battery*, *Field cleaned*, *Return to base*, and *Land*. For the tractors *Weed position*, *Tractor position*, *Move*, *Cut grass*, *Low battery*, and *Return to base*. Consequently, we placed tags in the points of the code where activities begin or finish. To better clarify the tag concept, Listing 1.1 shows an excerpt of the tag integration in the ROS code. Here the *Cut grass* function call is preceded and succeeded by publishing a topic containing the aforementioned fields.

```
tag = TagTopic()
tag.activity = 'CUT_GRASS'
tag.lifecycle = 'start'
ros_publisher.publish(tag)
CutGrass(ros_node)
tag.lifecycle = 'complete'
ros_publisher.publish(tag)
```

Listing 1.1. Excerpt of tag integration

Once we completed the tag integration, we ran the system in the Gazebo simulator. Our choice of using the simulation environment was mainly driven by the possibility of executing simulations with different initial settings, which allowed us to set random locations of weeds and variable battery levels for each robot.

[3] https://gazebosim.org/home.
[4] https://github.com/ros2/rosbag2.

To obtain a significant amount of data, we ran 70 simulations of the system. The execution of the simulations generates robotic event data that are processed to extract activity-centric event logs enriched with the spatial perspective.

Table 2 shows an excerpt of the multi-perspective event log related to a simulation run. The *Case* value uniquely identifies one run of the system execution, and the *Time* value stores a timestamp, i.e., the date and time in which the event occurred. *Activity* and *Lifecycle* are related to the high-level data generated via tags, and respectively represent the event name and its execution status. Considering the spatial perspective, the x, y, and z values store the position of the robot when an activity starts, is in progress, or completes. Finally, to distinguish the robot that generates a given event, its identifier is saved in the *Robot* field.

Table 2. Excerpt of multi-perspective event log

Case	Time	Activity	Lifecycle	Attributes			
				x	y	z	Robot
...	...	...	...	...	...	...	...
38	10:08:53.12	EXPLORE	inprogress	0.59	3.33	1.48	drone
38	10:08:55.01	EXPLORE	inprogress	0.99	4.01	1.48	drone
38	10:08:56.55	EXPLORE	inprogress	1.12	4.52	1.48	drone
38	10:08:57.61	EXPLORE	complete	1.66	5.03	1.48	drone
38	10:08:57.88	WEED_FOUND	start	1.66	5.03	1.48	drone
...	...	...	...	...	...	...	...
38	10:08:58.00	WEED_POSITION	complete	1.66	5.03	1.48	drone
38	10:08:58.30	WEED_POSITION	start	0.25	1.12	0.0	tractor_1
38	10:08:58.30	WEED_POSITION	complete	0.25	1.12	0.25	tractor_1
38	10:08:58.30	TRACTOR_POSITION	start	0.25	1.12	0.0	tractor_1
38	10:08:58.31	TRACTOR_POSITION	inprogress	0.25	1.12	0.0	tractor_1
38	10:08:58.32	TRACTOR_POSITION	complete	0.25	1.12	0.0	tractor_1
38	10:08:58.34	WEED_POSITION	start	0.17	1.12	0.0	tractor_2
38	10:08:58.35	WEED_POSITION	inprogress	0.17	1.12	0.0	tractor_2
38	10:08:58.36	WEED_POSITION	complete	0.17	1.12	0.0	tractor_2
38	10:08:58.37	TRACTOR_POSITION	start	0.17	2.04	0.0	tractor_2
38	10:08:58.38	TRACTOR_POSITION	inprogress	0.17	2.04	0.0	tractor_2
38	10:08:58.39	TRACTOR_POSITION	complete	0.17	2.04	0.0	tractor_2
38	10:09:00.91	CLOSEST_TRACTOR	start	1.66	5.03	1.48	drone
...	...	...	...	...	...	...	...

4.4 Analysis

We analyzed the produced multi-perspective event log both from the control-flow perspective and the spatial one. For this purpose, we developed and used a process mining tool that can perform the analysis and graphically show the resulting DFGs and the 3D plots. Notably, we present below the results of the analysis performed on the drone execution, whereas the analysis on the tractors' executions is in the online repository.

Control-Flow Analysis. Thanks to the event log produced by the TALE methodology we can analyze the mission of the robots using process mining.

Figure 3 represents the DFG discovered with the log prepared with TALE and depicts the process of the drone. Each node of the DFG represents the performed activity, and the arcs are labeled with the number of times the relation between the connected activities has been executed. The process shows that the drone properly performs the desired behavior, by starting with a *Take off* activity and then moving to the *Explore* activity. During this phase, it can navigate and look for weed grass. When a weed is identified, it triggers a *Weed found* and notifies the tractors through the *Weed position* message. Afterward, when it has received tractor responses, it moves to the *Tractor position* status. Then, it is able to elect the tractor closest to the weed, i.e., *Closest tractor*, and restart its exploration. Finally, the drone's execution ends when it spots a *Low battery* state. Therefore, it can perform the *Return to base* and *Land* activities. Notably, the extracted process shows that the drone has never performed the *Field cleaned* activity, meaning that it has not been able to explore the whole field.

Summing up, the control-flow analysis shows that the running robot worked as expected, without performing any incorrect behavior. However, to better understand the reason for not having an expected activity in the process, it is necessary to inspect other perspectives of the system that may have affected its behavior, such as space.

Spatial Analysis. The analysis of the space has been conducted through 3D charts that reflect the activities' spatial perspectives. First, the TALE tool extracts from the event log the robot's information related to its position in space. Figure 4 shows the space occupancy of the drone, in a 10×10 m field, during all 70 executions of the system. The chart shows that in none of the cases the drone's *Explore* activity, represented by red diamonds, has been able to navigate within the entire intended field. Specifically, we can see that the drone always takes off from the base station and starts the navigation of the field. During the navigation, it spots some weeds and reports them to the tractors. If, on the one hand, this sequence of actions has already been verified during the control-flow analysis, on the other hand, the 3D analysis of the space helps us to identify further potential errors in the system. Specifically, although the drone has properly performed the desired activities, it could not explore most of the field. This might be due to a programming error, which prevents the robot from navigating correctly. Additionally, a system developer may have overestimated the capabilities of the robots. For instance, a single drone may not be sufficient

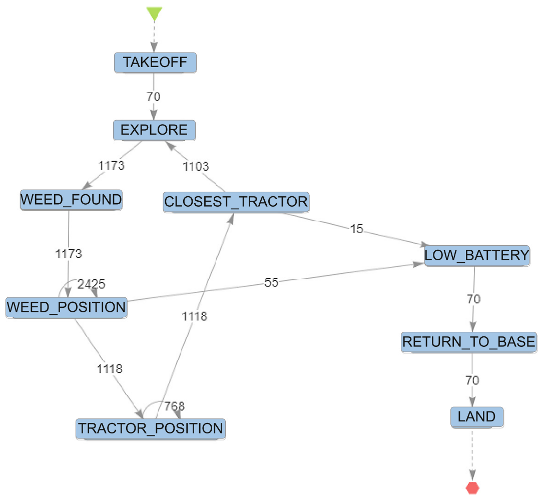


Fig. 3. Drone's DFG

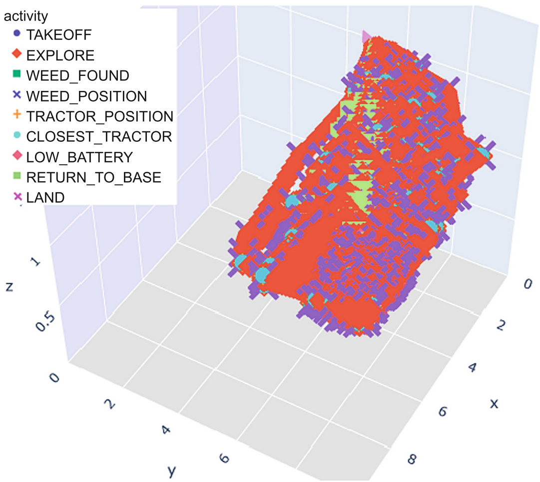


Fig. 4. 3D plot of 70 drone executions

for exploring the intended field, or the two tractors are too slow to satisfy the demands of the drone.

Moreover, exploiting the tool, we filtered the event log to analyze what happened in a single case. Figure 5 represents the TALE tool interface and the drone analysis during a selected case. Specifically, we can see that the drone takes off and starts exploring, during which it can spot 11 weeds. At one point, we can see that the drone has spotted a weed, but since it has not received a response, it continues to send periodic messages to trigger the tractors. During this idle

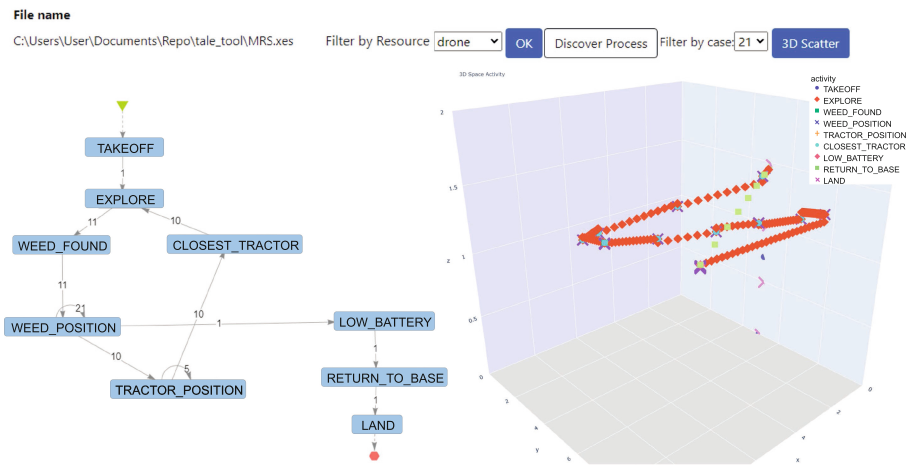


Fig. 5. TALE interface

state, the drone battery ran out, therefore it was properly able to come back to the base station and land. Indeed, this case highlights a problem in the robots' cooperation since a lack of a tractor response has led the drone to pause its navigation and send continuous messages, preventing it from identifying other weeds and fostering a battery drain.

5 Related Works

The literature collects different techniques and approaches for the analysis of robotic systems behavior [6,16]. The field test approach analyzes the real-world robots to check their behavior in different environment settings [20]. Differently, simulation environments can be used to analyze robots' behavior in a cheaper, safer, and more repeatable way [24]. The logging and playback approach collects logs during a system's operation for reproducing the whole system behavior [22]. The above-presented approaches do not allow a detailed analysis of robots' behavior since the user can visualize system execution, but cannot know in detail which activities led to a situation of interest. Moreover, analysis performed only via system visualization implies that the user checks continuously the execution status, thus requiring a large time consumption and making the analysis prone to human error. Different approaches exploit formal methods to specify, verify, and ensure the correctness of robotic systems [16]. However, the complexity of robotic systems leads to the well-known state-space explosion problem [27].

With respect to the process mining domain, recent works have applied techniques for the analysis and diagnosis of robotic systems [19,21]. Notably, despite the name similarity, the application of process mining to robots does not fall under the umbrella of Robotic Process Automation, i.e., a collection of tools that operate on digital systems by emulating human activities [3].

One of the most significant drawbacks faced by the presented works is due to the huge amount of data at a low-level of granularity generated by these systems. To solve the data granularity problem, the abstraction of high-level event logs from low-level ones is needed. Over the last few years, with the rise of IoT, many works focused on the abstraction of low-level events produced by sensors. Among the proposed approaches, the idea of labeling low-level event logs [17,25] is more effective in overcoming some limits generated by unsupervised learning techniques, such as identifying the name of clusters and choosing a custom level of abstraction [25]. However, labeled-based approaches aim to associate each low-level event with a label, to further extract high-level activities. Nevertheless, in robotic systems, a low-level event may have been used to perform different activities, thus it is unfeasible to label a low-level event statically. In this regard, the literature proposes a few approaches for enabling an expert to manually detect in the event log high-level activities starting from groups of, possibly shared, sensor data [23,26]. Differently from these approaches, TALE enables a domain expert to define the structure and the level of granularity of the event log before executing the system. In this way, the extracted logs properly reflect high-level activities and are suitable for the discovery of comprehensible process models.

Additionally, the enrichment of the event log with other system perspectives enables the development of new techniques for integrating the control-flow perspective with the spatial one. Indeed, the spatial perspective is marginally considered in process mining, since it is just exploited for automatically discovering humans' activities based on where they were performed [8]. However, the robotic system's behavior is directly influenced by the surrounding environment, thus leading to the increase of uncertainty in its behavior and the need to combine the analysis of its control-flow with the spatial perspective.

6 Conclusions and Future Works

In this paper, we presented the TALE methodology, which enables the use of process mining techniques to analyze multi-perspective robotic event logs. The methodology allows the developer of a robotic system to produce an event log that records data about the system execution and contains information from multiple perspectives. The obtained logs are activity-centric and suitable for process mining analysis. TALE currently prescribes a control-flow and spatial analysis that together guarantee an intuitive, yet effective, representation of robots' behavior. The methodology has been experimented on a robotic smart agriculture scenario using a tool we made available. The outcome of the performed analysis has shown the potentiality of process mining in the robotics field.

The strength of TALE lies in keeping the preparation separate from the analysis phase, using the event log as a unique point in common. Indeed, this separation makes it possible to extend each methodology step or even change one. For what concerns the preparation step, TALE uses ROS as the reference architecture for the robotic system and the log generation. However, the tagging approach can be embedded in any robot implementation up to some small adjustments. Moreover, the preparation can be extended to produce more sophisticated event logs containing other perspectives, for instance, a resource perspective reporting the robot's battery level, or a communication perspective where to report inter-robot messages. Regarding the analysis, this step strictly depends on the richness of the input event log. Indeed, each perspective can be analyzed with several process mining techniques, and additional perspectives can increase the effectiveness of the analysis [7]. For instance, the control-flow analysis, in particular the discovery, can be addressed with different discovery algorithms that also provide other process models [2]. Additionally, having other perspectives to analyze, the developer can correlate the activities with, for instance, message exchanges or domain-specific performance indicators.

In future work, we plan to improve the TALE preparation step by guiding the developer in the insertion of the tags. Secondly, we aim to improve the TALE analysis step by integrating performance indicators, such as the duration of the activities, into the process model visualization, and by providing other discovery algorithms targeting more advanced process model notations, e.g., BPMN. Moreover, this work has highlighted the need to relate the control-flow perspective with other aspects that are not typically addressed by process mining researchers, but that are essential to properly analyze robotic systems, such as the spatial perspective. Indeed, we aim to exploit emergent research topics that aim to integrate the control-flow perspective with other aspects of the system, such as the one in [14]. Furthermore, traditional process mining techniques focus on logs generated by a single entity, while in a multi-robot system, the logs come from multiple robots. Thus, we plan to investigate process mining approaches specifically designed for distributed and collaborative scenarios, such as the one proposed in [13], for discovering the interactions between the robots.

References

1. van der Aalst, W.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.: Foundations of process discovery. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 37–75. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_2
3. Van der Aalst, W., Bichler, M., Heinzl, A.: Robotic process automation. Bus. Inf. Syst. Eng. **60**, 269–272 (2018)
4. Afanasyev, I., et al.: Towards the internet of robotic things: analysis, architecture, components and challenges. In: DeSE, pp. 3–8. IEEE (2019)
5. Afyouni, I., Ray, C., Claramunt, C.: Spatial models for context-aware indoor navigation systems: a survey. J. Spat. Inf. Sci. **4**, 85–123 (2012)
6. Afzal, A., Goues, C.L., Hilton, M., Timperley, C.S.: A study on challenges of testing robotic systems. In: ICST, pp. 96–107. IEEE (2020)
7. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. IEEE TKDE **31**(4), 686–705 (2018)
8. Bertrand, Y., Van den Abbeele, B., Veneruso, S., Leotta, F., Mecella, M., Serral Asensio, E.: A survey on the application of process mining to smart spaces data.

In: Montali, M., Senderovich, A., Weidlich, M. (eds.) ICPM 2022. LNBIP, vol. 468, pp. 57–70. Springer, Cham (2022)
9. Bertrand, Y., De Weerdt, J., Serral, E.: A bridging model for process mining and IoT. In: Munoz-Gama, J., Lu, X. (eds.) ICPM 2021. LNBIP, vol. 433, pp. 98–110. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98581-3_8
10. Bourr, K., Corradini, F., Pettinari, S., Re, B., Rossi, L., Tiezzi, F.: Disciplined use of BPMN for mission modeling of Multi-Robot Systems. In: Forum at Practice of Enterprise Modeling, vol. 3045, pp. 1–10. CEUR-WS.org (2021)
11. Chitic, S.: Middleware and programming models for multi-robot systems. Ph.D. thesis, INSA de Lyon, France (2018)
12. Corradini, F., Pettinari, S., Re, B., Rossi, L., Tiezzi, F.: A BPMN-driven framework for Multi-Robot System development. Robot. Auton. Syst. **160**, 104322 (2023)
13. Corradini, F., Re, B., Rossi, L., Tiezzi, F.: A technique for collaboration discovery. In: Augusto, A., Gill, A., Bork, D., Nurcan, S., Reinhartz-Berger, I., Schmidt, R. (eds.) BPMDS. LNBIP, vol. 450, pp. 63–78. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07475-2_5
14. Fahland, D.: Process mining over multiple behavioral dimensions with event knowledge graphs. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 274–319. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_9
15. Kortenkamp, D., Simmons, R., Brugali, D.: Robotic systems architectures and programming. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 283–306. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1_12
16. Luckcuck, M., Farrell, M., Dennis, L.A., Dixon, C., Fisher, M.: Formal specification and verification of autonomous robotic systems: a survey. ACM Comput. Surv. **52**(5), 100:1–100:41 (2019)
17. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 125–141. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_8
18. zur Muehlen, M., Swenson, K.D.: BPAF: a standard for the interchange of process analytics data. In: zur Muehlen, M., Su, J. (eds.) BPM 2010. LNBIP, vol. 66, pp. 170–181. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20511-8_15
19. Nicoleta, T.C.: Process mining on a robotic mechanism. In: ICST Workshops, pp. 205–212. IEEE (2021)
20. Otsu, K., et al.: Supervised autonomy for communication-degraded subterranean exploration by a robot team. In: Aerospace Conference, pp. 1–9. IEEE (2020)
21. Roldán, J.J., Olivares-Méndez, M.A., del Cerro, J., Barrientos, A.: Analyzing and improving multi-robot missions by using process mining. Auton. Robot. **42**(6), 1187–1205 (2018)
22. Schmidt, M., Kirchhoff, J., von Stryk, O.: A modular and portable black box recorder for increased transparency of autonomous service robots. Robot. Autom. Lett. **7**(4), 10673–10680 (2022)
23. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from IoT data. Future Internet **15**(2), 77 (2023)
24. Takaya, K., Asai, T., Kroumov, V., Smarandache, F.: Simulation environment for mobile robots testing using ROS and Gazebo. In: ICSTCC, pp. 96–101. IEEE (2016)

25. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
26. Van Eck, M.L., Sidorova, N., Van der Aalst, W.: Enabling process mining on sensor data from smart products. In: International Conference on Research Challenges in Information Science, pp. 1–12. IEEE (2016)
27. Vicentini, F., Askarpour, M., Rossi, M.G., Mandrioli, D.: Safety assessment of collaborative robotics through automated formal verification. IEEE Trans. Robot. **36**(1), 42–61 (2020)
28. Weerdt, J.D., Wynn, M.T.: Foundations of process event data. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 193–211. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_6
29. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granul. Comput. **6**(3), 719–736 (2021)

An Approach for Face Validity Assessment of Agent-Based Simulation Models Through Outlier Detection with Process Mining

Rob Bemthuis[1,2](✉) and Sanja Lazarova-Molnar[2,3]

[1] University of Twente, Enschede, The Netherlands
r.h.bemthuis@utwente.nl
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
rob.bemthuis@partner.kit.edu, sanja.lazarova-molnar@kit.edu
[3] University of Southern Denmark, Odense, Denmark
slmo@mmmi.sdu.dk

Abstract. When designing simulations, the objective is to create a representation of a real-world system or process to understand, analyze, predict, or improve its behavior. Typically, the first step in assessing the credibility of a simulation model for its intended purpose involves conducting a face validity check. This entails a subjective assessment by individuals knowledgeable about the system to determine if the model appears plausible. The emerging field of process mining can aid in the face validity assessment process by extracting process models and insights from event logs generated by the system being simulated. Process mining techniques, combined with the visual representation of discovered process models, offer a novel approach for experts to evaluate the validity and behavior of simulation models. In this context, outliers can play a key role in evaluating the face validity of simulation models by drawing attention to unusual behaviors that can either raise doubts about or reinforce the model's credibility in capturing the full range of behaviors present in the real world. Outliers can provide valuable information that can help identify concerns, prompt improvements, and ultimately enhance the validity of the simulation model. In this paper, we propose an approach that uses process mining techniques to detect outlier behaviors in agent-based simulation models with the aim of utilizing this information for evaluating face validity of simulation models. We illustrate our approach using the Schelling segregation model.

Keywords: Face validity · Agent-based simulation · Process mining

1 Introduction

Face validity is a key aspect of simulation model design. It involves asking knowledgeable individuals about the system if the model and its behavior are plausible

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 134–151, 2024.
https://doi.org/10.1007/978-3-031-46587-1_8

[24]. This step is commonly used to evaluate how well the model captures the essential features of the real-world system that it represents [16,25]. Face validity methods rely on human expertise and include expert assessments and structured walk-throughs [15]. This process ensures that the simulation model outcomes are reasonable and plausible within the theoretical framework and implicit knowledge of system experts or stakeholders. Although other types of validation, such as statistical validation, are also important, face validity is relevant because it is a common first step in assessing the simulation model's validity. Nevertheless, after designing their model, practitioners and researchers should rigorously test its performance under a variety of conditions [9].

Assessing face validity in simulation modeling can be challenging [20]. One major challenge is the potential for a discrepancy between the model's assumptions and the real-world system it represents, which can affect the accuracy of the model [16]. The complexity of the modeled system can also make achieving face validity challenging, as can balancing model accuracy with simplicity [9]. Additionally, the subjective nature of face validity and the potential for biases in the validation process must be taken into account [28]. Incorporating feedback from subject matter experts and stakeholders can enhance face validity, but this process can be time-consuming and resource-intensive. For instance, if assessors find it difficult to categorize and examine every available option, achieving comprehensive face validity can become challenging [10].

To this end, the emerging field of process mining can provide a valuable tool for enhancing face validity in simulation models and addressing some of the concerns previously mentioned. Process mining extracts knowledge from event logs of real-life processes [1] and can validate behavior in simulation models against real-world behavior. By comparing simulation output with data extracted from real-world processes, process mining can help identify discrepancies. This allows for adjustments to be made to the simulation model to improve its validity, as discussed in Subsect. 2.2. The wealth of techniques developed within the process mining discipline can also be applied to event logs generated by a simulation model. This enables experts in simulation and process mining to assess resulting process models and corresponding performance insights for face validity. By utilizing process mining, including its visually appealing discovered process models, simulation modelers can provide a novel way to ensure that simulated processes and outcomes are consistent with real-world systems. This can contribute positively to performing face validity assessments.

In simulation models, outliers can significantly impact the model's validity. Outliers, which lie an abnormal distance from other values (in an arbitrary sample), can draw attention to unusual behavior, either raising doubts about or reinforcing the model's credibility. Outliers also play a key role in conducting face validity checks. By identifying and addressing outliers, the accuracy and reliability of a simulation model can be enhanced, leading to more valid conclusions and improved decision-making. Outliers can reveal important and unexpected behaviors that capture a wide range of real-world phenomena. By verifying that certain cases are indeed outliers, the model's face validity can be strengthened. For instance, in a disease spread simulation model, the identification of an outlier case as a 'super-spreader' can bolster the model's face validity by accurately representing the significant impact of super-spreaders on disease spread in real-world scenarios. Conversely, outliers may also compromise the validity of a simulation model if their existence is doubted or considered unrealistic by experts in the field.

In this paper, we propose an approach that employs process mining techniques to detect outlier behaviors in agent-based simulation (ABS) models, with the goal of enhancing the face validity assessment process. While face validity is generally essential for many types of simulation, it is particularly important for ABS [15]. The distinctive attributes of ABS models, including the representation of heterogeneous agents and the emphasis on emergent behavior, underscore the criticality of emphasizing face validity in this domain. Our study makes two main contributions: (1) we apply process mining techniques to extract and identify outlier behaviors from an ABS model, and (2) we incorporate human expertise to conduct a face validity assessment on the knowledge extracted through process mining, thereby reinforcing human judgement in the evaluation process. We demonstrate the versatility and potential usefulness of our approach using the Schelling model of segregation, a popular ABS model, and show how it can be applied to various scenarios of the ABS model. Our approach is guided by Peffers' Design Science Research Methodology [21], as reflected in the structure of this article.

The remainder of this paper is organized as follows. Section 2 describes background on face validity for ABS models and how process mining can enhance the face validity of ABS modeling practices. Section 3 presents our approach. Section 4 demonstrates our approach through an illustrative scenario. Finally, Sect. 5 positions related work and Sect. 6 concludes and provides pointers for further work.

2 Background

In this section, we begin by providing a concise overview of face validity techniques for ABS models. Subsequently, we explore the potential benefits of incorporating process mining techniques into the face validity process of ABS models.

2.1 Face Validity Techniques for Agent-Based Simulation Models

Model validation is the process of assessing the accuracy of a simulation model in representing a real-world system with respect to the study objectives [3] (see also Fig. 1). This involves comparing the model's output to empirical data, experimental results, or expert knowledge to ensure that it realistically represents the system being modeled. Face validity is one specific validation technique, which is considered to be relatively informal and subjective [24]. It involves soliciting feedback from individuals who are knowledgeable about the system being

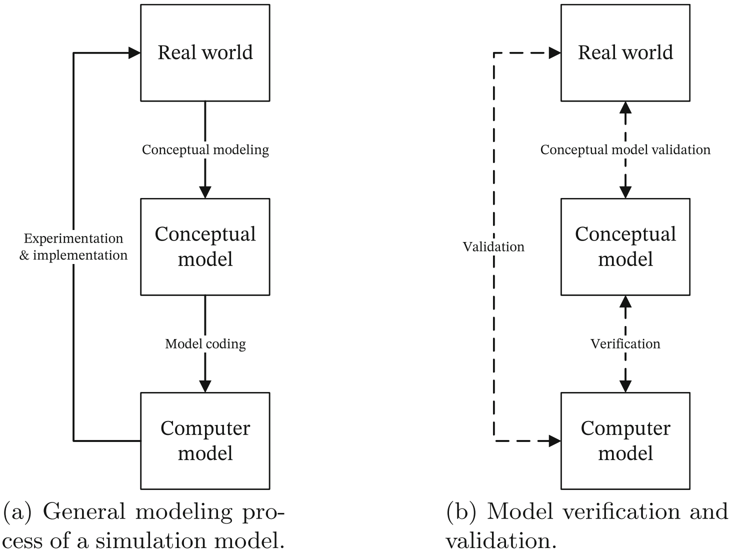


Fig. 1. Verification and validation of a simulation model (adapted from [22]).

modeled to determine whether the model and its behavior are reasonable and realistic.

Several face validity techniques have been proposed in the literature, and these techniques are not necessarily mutually exclusive. Table 1 provides an overview of these techniques. Ideally, these techniques should be conducted by independent groups of human experts [15]. While there are other face validity techniques in the field of simulation modeling, our study focuses specifically on techniques related to ABS models. Focusing on ABS models narrows the scope and limitations of our research to a more specific domain.

The methodology for conducting a rigorous face validity assessment of a simulation model is subject to debate and may depend on various factors. However, several general guidelines can be proposed. Firstly, evaluators must possess *sufficient knowledge and expertise* about the system being modeled. Secondly, the model must be *transparent and comprehensible* to the evaluators. It should have explicit explanations of its assumptions, inputs, and outputs. Thirdly, evaluators should be presented with *realistic scenarios* that accurately reflect the system's expected behavior. Finally, face validity assessment should be considered an *iterative process*, with modifications made to the model based on feedback received from the evaluators.

2.2 Achieving Face Validity Through Process Mining Techniques

Process mining provides a data-driven approach to supplementing the face validity process of simulation models. It facilitates a comparison between real-world processes and the behavior of the simulation model (see Fig. 2). By extracting

Table 1. Face validity techniques for ABS models reported in literature.

Name	Reference(s)	Description
Animation	[15, 28]	Graphical display of model behavior over time
Output assessment	[15]	Check the plausibility of absolute values, relations, dynamics, and trends of output values in simulation runs
Immersive assessment	[15]	Evaluate the behavior of an isolated simulated agent by observing its perceptions and reactions through its interface. The expert can also assess the behavior of other agents by interacting with the human-controlled agent if the interface allows participation
Turing test	[9, 28]	Test if experts distinguish between model-generated and real-world data
Graphical representation	[28]	Visualize the model's output data with graphs
Tracing	[28]	Isolated entities in the model are monitored for their behavior
Internal validity	[28]	Compare the results of multiple replications of a stochastic simulation model using different random seeds. Inconsistent sample points resulting from the random number generators indicate issues with either the programming model or the conceptual model
Historical data validation	[28]	When historical data is available, the model is built using a portion of the data (training sets), and the remaining data (test sets) is used to verify whether the model emulates the behavior of the system
Sensitivity analysis	[28]	Model input and parameters are adjusted to examine their impact on the model's behavior and input, with the expectation that the model will reflect the real system. Sensitive parameters, which significantly affect the model's behavior, must be accurately determined before the model can be utilized
Predictive validation	[28]	Compare the model's predictions to actual system behavior, which can be obtained from operational systems or experiments, including laboratory or field tests

information about the actual execution of processes from event logs generated by both real-life processes and simulation models, process mining techniques can discover process models that graphically represent the flow of activities in a chart (e.g., through nodes, activities, and gateways). The discovered process models and performance insights, such as throughput and waiting times, can provide valuable insights into the execution of activities. For instance, comparing the extracted process model with the expected real-life behavior as determined by experts can enhance the validity of a simulation model by identifying discrepancies. Process mining can provide a systematic approach for analyzing data and identifying outliers or deviations from expected patterns.

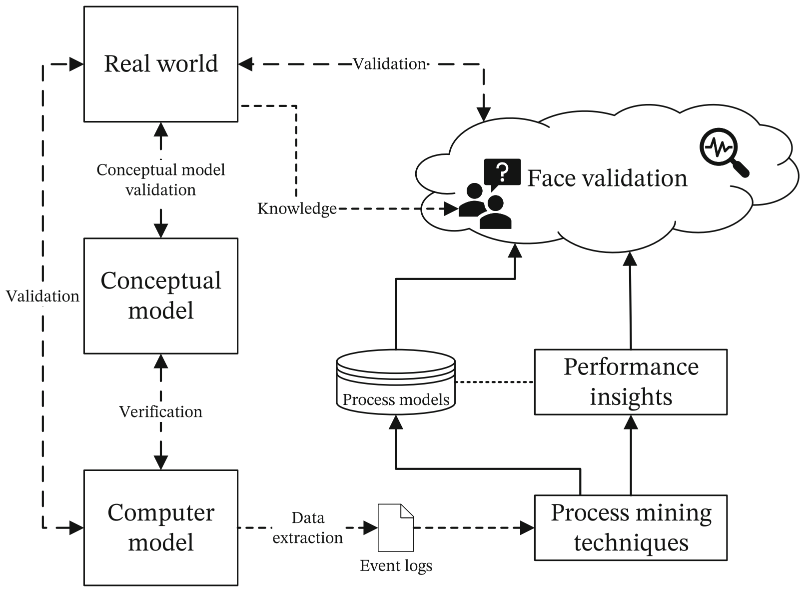


Fig. 2. Visual representation of using process mining techniques to assess the face validity of a simulation model.

To our knowledge, there is limited research on using process mining to enhance the face validity of ABS models. Although some studies may implicitly employ process mining methods, few explicitly discuss their use in the context of face validity or related terms (see Sect. 5) such as plausibility checks. Nonetheless, we believe that process mining techniques can serve as a valuable tool for assessing the face validity of simulation models, as per the guidelines outlined in the preceding subsection. Firstly, assessors of agent-based systems are expected to possess domain knowledge about the system being modeled and its Key Performance Indicators (KPIs). Process mining techniques can provide insights into a wide range of performance indicators that align with the modeled system's KPIs. Secondly, the results obtained through process mining are based on the analysis of event logs, allowing users to trace the flow of events and understand how the results were obtained. This is important for accurately representing emergent behavior, interaction dynamics, and outlier behaviors in ABS models. Furthermore, visually engaging mined process models can enhance usability and comprehensibility among individuals who lack specialized expertise in the field of process mining but are familiar with the agent system being modeled [6]. Thirdly, evaluators can be presented with specific scenarios of preference or interest (e.g., outlier behaviors) based on event records that describe the actual functioning of the system due to the granularity and sophistication of process mining techniques and many available tools (e.g., for filtering traces) [23]. Finally, process mining can be used iteratively to enhance face validity by regularly comparing the simulation model's behavior with real-world processes. For example, real-time streaming of event data, combined with validity checks at set intervals, allows for monitoring of the model's performance and identification of exceptional behaviors. Injecting new event logs provides additional data points for testing and refining model performance, thereby enhancing accuracy.

3 An Approach for Assessing Face Validity

In this section, we present an approach for assessing face validity of ABS models. We first provide an overview of the approach and then discuss each step of the approach in detail. Our approach is illustrated in Fig. 3 and comprises six steps that leverage execution logs extracted from an ABS model.

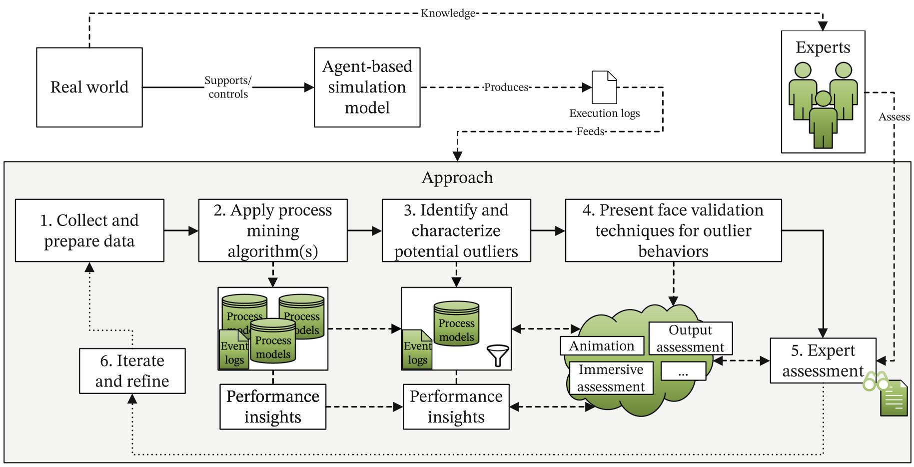


Fig. 3. A face validity assessment approach of ABS models using process mining.

Step 1: Collect and Prepare Data. The initial step involves selecting, cleaning, and aggregating (event) log files from the real-world system being modeled. These files are then prepared for analysis using process mining techniques, which entails locating the raw data, migrating and transforming it into an event log file format, and pre-processing the data (e.g., filtering and cleaning).

Step 2: Apply Process Mining Algorithm(s). The next step involves using process mining techniques to extract information about the actual execution of processes from the event logs. In this step, we select appropriate process mining discovery techniques to mine process models and assess their quality. This assessment is important as multiple process models may be generated, and their representation quality (e.g., fitness and precision) and other quality indicators (e.g., number of event logs) must be considered.

Step 3: Identify and Characterize Potential Outliers. The third step entails ensuring that the obtained process model and insights can be used to identify potential outliers. This can be achieved by displaying different abstractions of the process model through filtering to examine specific aspects of the process. Additionally, the process model can be augmented with various process metrics (i.e., KPIs), such as frequency metrics (e.g., absolute and case frequency, maximum number of repetitions), performance metrics (e.g., total duration, minimum/median/maximum duration), or combined metrics. The identified outliers can then be characterized to understand their deviation from the norm. This involves analyzing their attributes and features to determine their uniqueness or difference, which may involve examining the attributes of the process, the context of the outlier, and potential contributing factors.

Step 4: Present Face Validation Techniques for Outlier Behaviors. The fourth step involves selecting, configuring, and presenting appropriate face validation techniques for use by human assessors, such as visual inspection of the model's behavior and solicitation of feedback from domain experts. Table 1 presents additional examples of face validity techniques. The specific techniques chosen depend on the nature of the ABS and process insights, as well as the availability of relevant data and expertise. Individuals familiar with the system being studied should be instructed on the use of the face validity procedures.

Step 5: Expert Assessment. The fifth step involves having experts assess whether the behaviors presented through the use of process mining techniques are plausible. These experts can identify any discrepancies between the simulation model and the real-world system, or confirm the accuracy of the model's representation of system behaviors.

Step 6: Iterate and Refine. After conducting a face validity check and establishing sufficient credibility in the ABS model, it is common to proceed with other validation assessments, such as quantitative statistical analysis. However, adjustments may be made to the simulation model to more accurately represent reality and followed by additional face validity assessments. This iterative process involves assessing the simulation model's behavior against real-world processes and making modifications based on feedback from the face validity assessment. This can gradually improve the accuracy and reliability of the ABS model, leading to more valid conclusions and better decision-making. The iteration process can also involve refining previous steps, such as addressing new potential outlier behaviors based on feedback from human assessors and involving different or new experts in conducting face validity assessments.

4 Illustrative Scenario

In this section, we present an illustrative scenario demonstrating the application of our approach using the Schelling model of segregation. We introduce the scenario and detail the key actions taken to apply our approach. Selected outcomes are presented for clarity and conciseness, with a focus on the illustrative nature of the scenario. We conclude by summarizing the lessons learned and the challenges encountered.

4.1 The Schelling Model of Segregation

The Schelling model of segregation is a social simulation model that demonstrates how individual preferences can lead to large-scale social patterns, even with low levels of discrimination or prejudice [26]. The model has been applied in various research fields, has inspired policy-makers and planners to develop strategies for promoting diversity and reducing segregation in urban areas, and has also served as a basis for developing other simulation models exploring social phenomena.

In the classic Schelling segregation model, a grid representing a housing market is filled with individuals who possess a "tolerance threshold" for the percentage of their neighbors that must be of the same race or ethnicity [26]. As the simulation progresses, individuals who are not satisfied with their neighbors relocate to new positions on the grid that meet their tolerance threshold. This can result in the formation of highly segregated neighborhoods as individuals with similar traits congregate. This congregation can occur even when individual preferences are not extreme but rather moderate.

Conducting a face validation assessment through process mining for the Schelling model of segregation is relevant for several reasons. First, process mining techniques can help identify and characterize outliers in the simulation model that may undermine or fortify its validity. By addressing these outliers, the accuracy and reliability of the model can be improved, which can lead to more valid conclusions and better decision-making. Second, the Schelling model of segregation is a widely recognized and influential social simulation model that has been applied in various fields of research. Ensuring its face validity is important for maintaining its credibility and usefulness as a tool for understanding complex social phenomena and for the wider (agent-based) simulation modeling community.

4.2 Demonstration of the Approach

Step 1: Collect and Prepare Data. For our ABS model, we used the Schelling model implementation described by [7], which we modified using Python 3.6.9 and the AgentPy 0.1.5 library [11]. We limited extraction to the event logs of the scenarios presented in Table 2. These scenarios cover a variety of situations and differentiate among several model parameters. "Ruleset type" refers to the model's configuration where either all agent groups have the same tolerance threshold (homogeneous population) or all but one agent group have the same tolerance threshold (heterogeneous population).

Table 2. Scenarios of the Schelling model considered for event log extraction.

Scenario	Density	Grid size	Ruleset type	Tolerance threshold (%)
1	0.80	20×20	homogeneous	0.55
2	0.70	20×20	homogeneous	0.55
3	0.70	20×20	homogeneous	0.20
4	0.70	20×20	heterogeneous	0.10

Table 3 provides an example of a produced event log. The activities included three types: *moveLocation* (i.e., an agent moves from one location to another), *changeHappy* (an agent's status changes from happy or unhappy to happy), and *changeUnhappy* (an agent's status changes from happy or unhappy to unhappy). For the timestamp, we adopted a similar approach to that described by [7]. We assigned a sequential counter to each step in the model's execution based on the order in which it occurred chronologically.

Table 3. An excerpt of an event log generated.

timestamp	counter	activity	caseID	coordinates	directNeighbors	happinessLevel
2022-01-01 12:31:05	0	changeHappy	23	18,5	2, 14, 22, 87	0.75
2022-01-06 09:00:00	0	moveLocation	55	6,3	13, 16, 56, 61, 81	0.84
2022-01-06 09:00:00	1	changeHappy	13	5,2	8,16,41,55,56,77,83	0.90
2022-01-06 09:00:00	2	changeUnhappy	16	6,2	3,13,55,56,77,81	0.45
2022-01-06 09:00:00	3	changeUnhappy	56	5,3	8,13,16,41,55,61	0.29
2022-01-06 09:00:00	4	changeHappy	61	6,4	6,31,55,56,73,81	0.66
2022-01-06 09:00:00	5	changeHappy	81	7,3	2,16,55,61	0.78
2022-02-01 10:23:00	0	moveLocation	33	11,16	5,25	0.81

For analysis purposes, we concatenated the naming convention of an activity to include both the number of neighbor agents and the number of neighboring agents of a similar group (i.e., *changeHappy_X_Y*, where X = number of neighbors and Y = number of neighbors of a similar group). This naming convention ensured that there was sufficient data to obtain a realistic overview of multiple scenarios while keeping the entire process manageable in size.

Step 2: Apply Process Mining Algorithm(s). When conducting a face validity assessment based on event logs produced by an agent-based system, the choice of process mining discovery algorithm depends on the specific characteristics of the data and the desired outcome. For instance, if the data contains a significant amount of noise or if the process being modeled is less structured, then the Fuzzy Miner might be a more suitable choice [13]. However, if the data is relatively clean and well-structured, then the Heuristic Miner might be more appropriate [12].

We chose the Fuzzy Miner for its efficiency and ease of use with less structured processes. By using significance/correlation metrics to simplify the process model, it provides reliable results for complex data sets [13]. It can also exclude or cluster less important activities. The Fuzzy Miner can animate the event log on top of the created model, providing an understanding of dynamic process behavior, which is desirable for assessing face validity. We selected Disco as our process mining tool due to its usability, fidelity, and performance [14].

Step 3: Identify and Characterize Potential Outliers. We employed process mining techniques to filter the representations and present outlier behaviors. As an illustration, we demonstrate how animation, output, and immersive assessments (as described by [15]) can be conducted. For the identification and characterization of outliers, we analyzed the attributes and features of (potential) outlier behaviors. Further details and visual representations are provided in the following step.

Step 4: Present Face Validation Techniques for Outlier Behaviors. We demonstrate the application of three face validity techniques: animation, output, and immersive assessment. Due to space constraints, we present only a selection of these outcomes. In the following, we provide examples of the information that could be presented to an individual tasked with assessing validity. The next step presents an example of an expert assessment.

Animation Assessment: We created a graphical representation that displays the progress of the graphical model behavior over time in distinct time intervals. All paths and activities were shown, allowing the assessor to visually observe exceptions and identify possible deviations. Figure 4 shows a snapshot of the animation shown for scenario 2. The evaluator can examine various behaviors from the animation and focus on outliers based on attributes such as color and arrow thickness. This animation is particularly useful for intuitively identifying and highlighting bottlenecks in the process. When numerous cases accumulate on a specific arc, causing congestion, the tool aggregates these cases into larger "bubbles", emphasizing bottlenecks in the process.

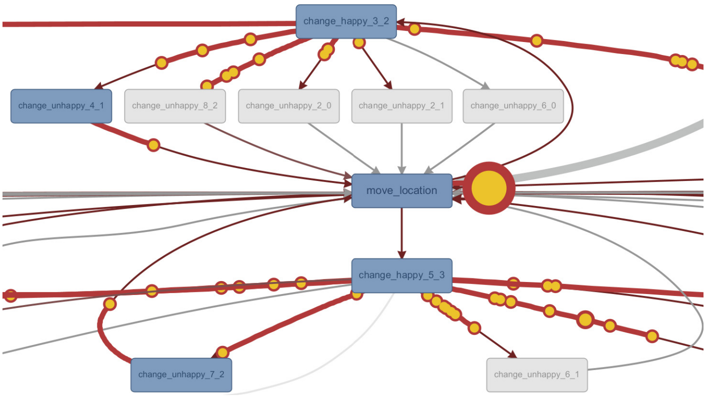


Fig. 4. Snapshot of the graphical animation used for scenario 2.

Output Assessment: We generated a graphical representation of the case duration (i.e., steps conducted by an agent), supplemented by additional statistics such as the number of events, cases, and activities for agents that exhibited movement at least once. Figure 5 illustrates an example of the visual representation employed for scenario 2. This visual representation enables experts to conduct an initial assessment of the plausibility of the ABS model. Using this example, the expert can evaluate extreme cases, such as case durations with the highest number of cases, or identify trends.

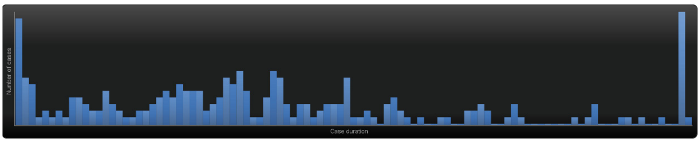


Fig. 5. Example representation used for output assessment of scenario 3.

Immersive Assessment: We identified agents that deviated from the average behavior by analyzing cases with an exceptionally high number of associated events, indicating unusual frequency in their movement patterns. A detailed overview of each agent's activities, including timestamps and happiness status, was provided to the human assessor for evaluation. Since activity names encode information about both the number of neighbors and the number of neighbors belonging to a similar group, this information could provide valuable insights into the validity of the ABS model. Tables 4 and 5 present examples of an agent's trace.

Table 4. An excerpt of traces used for immersive assessment for scenario 1, case 206 with 112 events.

Activity	Time	happinessLevel
...	...	...
change_happy_4_3	01:01:17	false
change_unhappy_7_3	01:01:19	true
move_location	01:01:20	false
move_location	01:01:21	false
change_happy_5_3	01:01:21	false
change_unhappy_2_0	01:01:22	true
...	...	...

Table 5. An excerpt of traces used for immersive assessment for scenario 4, case 248 with 6 events.

Activity	Time	happinessLevel
move_location	01:00:01	false
move_location	01:00:02	false
move_location	01:00:03	false
move_location	01:00:04	false
move_location	01:00:05	false
change_happy_8_3	01:00:05	false

Step 5: Expert Assessment. We provided an expert with a template that included an introduction to the ABS model, its purpose, the assessment method, and instructions for evaluation. The evaluation criteria were listed, and space was provided for the expert to record their observations and provide feedback on the face validity of the ABS model's outcome. The evaluation criteria included: (1) accurate representation of the real-world system being modeled; (2) consistency of agent behavior with their real-world counterparts; and (3) plausibility of the overall simulation model outcome. A human assessor evaluated the face validity based on the artifacts discussed in step 4. The expert evaluated the face validity and scored it as either 'plausible' or 'not plausible'. The assessment results and conclusions regarding the face validity of the ABS model's outcome were summarized, and selected outcomes are reported below.

During both the animation and immersive assessments, an observation was made regarding the perceived unrealistic movement time of agents from one location to another. While this can be attributed to model limitations as described by [7], it is not considered plausible in reality. Another concern, as discovered during the animation assessment, was the simultaneous movement of multiple agents at similar time steps. While this may be attributed to exceptional real-life behaviors or administrative practices in the housing market (e.g., movements designated as "official" at the start of a new day), its repeated occurrence during the animation led the human assessor to deem this model behavior not plausible in reality.

During the output assessment of scenario 3 (Fig. 5), the human assessor observed two distinct peaks. The first peak occurred at the beginning when

16 cases were satisfied after a single move, possibly due to dissatisfaction with a single neighbor. The second peak involved 17 cases that continued to move after 100 time units, representing approximately 6% of the total population. This observation raised concerns about the validity of the ABS model and warrants further investigation, as it was perceived as not plausible in reality.

In addition to the previously mentioned unrealistic timing of events, the immersive assessment yielded mixed results regarding the model's validity as determined by the assessor. For example, the event log trace of case 206 in scenario 1 (as depicted in Table 4) indicates a not plausible number of agent movements, while the observed pattern of transitioning between happy and unhappy states was deemed consistent with real-world practices.

Step 6: Iterate and Refine. Currently, this step is still in progress. Further examination is needed to thoroughly explore the discrepancies. Discussions are also underway about implementing our approach on a real-world dataset and involving policy-makers, real estate agencies, homeowners or renters in the panel of experts to validate the Schelling model.

4.3 Discussion

In this subsection, we briefly discuss the lessons learned and open challenges encountered during our case study. Our findings suggest that the outcomes obtained through process mining techniques were intuitive and easy to follow, making the assessment process relatively efficient. However, we also identified a need for more structured approaches to conducting face validity assessments of ABS models using process mining techniques.

One challenge we encountered was matching the KPIs used in practice with those obtainable through process mining techniques. Further research is needed to address this issue and ensure that KPIs used in face validity assessments are appropriate and well-known to experts. Another open research question concerns addressing the adaptive or changing behavior of agents in ABS models, as also highlighted by [5,6]. Concept drift incorporation into face validity assessments is also important. Further research is needed to develop methods for incorporating dynamic aspects of ABS models, such as their evolving nature, heterogeneity, and complex interactions, into face validity assessments.

We evaluated our proposed method using a segregation model across four scenarios. Although a comprehensive presentation of results for each scenario based on established criteria could provide valuable insights, we have chosen to emphasize only specific findings in this paper due to resource and time constraints as well as limited publication space. It is important to note that our evaluation was limited by our reliance on a single expert opinion for behavior assessment and the application example used in our study may not be representative enough to make more fundamental statements.

5 Related Work

Previous research highlights the importance of face validity in enhancing ABS model credibility. Several methods, including expert panels and stakeholder engagement (see e.g., Table 1), have been proposed for assessing face validity. The benefits of incorporating face validity assessments in ABS model design and implementation for decision-making and policy development are also emphasized.

For example, [27] developed a computational model of collaborative learning health systems using an agent-based approach and demonstrated its initial computational and face validity. The authors demonstrated face validity by examining the effects of varying a single parameter. However, they acknowledge the model's face validity for a wide range of stakeholders is unknown and call for further refinement through collaboration with experts. In other work, [2] proposed a framework for evaluating health care markets through agent-based modeling and presented a face-validity assessment procedure by examining the degree to which empirical studies support key theorized relationships within health care markets and comparing them with relationships generated by their model. Furthermore, [18] presented an agent-based model of a stock market that incorporates common-sense evidence and implements realistic trading strategies based on practitioners' literature. The model was validated using a four-step approach consisting of face validity assessment, sensitivity analysis, calibration, and validation of model outputs.

In the process mining domain, [17] used semi-structured interviews to evaluate the face validity of process mining results, but the specific questions used during the interviews were not reported. In the ABS domain, [8] used numerical simulations to test the face validity of a part of their ABS model. While the authors' findings support the plausibility of the outcomes within the theoretical framework, they advocate for further empirical estimation of model parameters through real-world measurements. Work described in [19] used feedback from a project manager to assess the face validity of their model. In [4], the authors conducted an exploratory study on face validity assessment in ABS models, presenting a proof-of-concept that combines process mining with visualization. Their results provide initial evidence of the effectiveness of this approach, but also highlight the need for further research to gain a finer-grained understanding of agent-level dynamics, such as studying emergent behaviors at specific group levels, including outlier behaviors.

Our work builds upon previous research by applying existing process mining techniques to identify outliers in an ABS model and having a human expert assess these outliers. Previous approaches focused on ensuring that KPIs, such as average cycle time or work-in-progress levels, aligned with observed KPIs, but neglected to investigate outlier behavior. By including this behavior, we provide meaningful insights into the validity of an ABS model. Furthermore, by incorporating process mining techniques into the design and execution of ABS models, we enable face validity assessments even for non-experts. Our proposed approach, combined with the application of the Schelling model of segregation, represents an initial step towards analyzing complex socio-technical systems typically modeled and simulated using agent-based techniques.

6 Conclusion and Future Work

In this paper, we presented an approach using process mining techniques to detect outliers in ABS models and evaluate their face validity. Our approach leverages human expertise to perform a face validity assessment on the information obtained from the process mining analysis, focusing on outlier behaviors. We demonstrated our approach using the Schelling model of segregation and showed how it can be used to assess face validity. In particular, we demonstrated how animation, output assessment, and immersive assessment can be employed as face validity techniques for ABS models. This study offers valuable insights for enhancing the face validity assessment process of ABS models using process mining and holds broader potential for advancing the field of simulation modeling.

Our study findings are subject to validity threats due to the complexity of ABS models and the need for specialized face validation techniques to accurately capture agent behavior and emergence. We focused on a small set of techniques and demonstrated our approach in a limited experimental environment, limiting the generalizability of our results. Face validity is subjective and reliant on the validator's judgment, making it challenging to standardize or quantify results and potentially leading to inconsistencies in the validation process. Additionally, our inspection of a restricted subset of simulation outputs may fail to capture essential aspects of the simulation's behavior, potentially resulting in an incomplete evaluation of the model's validity. Nevertheless, ABS models have unique characteristics such as complex agent interactions, stochasticity, and emergent behaviors, making them challenging to comprehensively assess by experts.

Future work will involve refining and extending our approach by adapting ABS models to various domains and settings, investigating a broader range of process mining techniques, and including insights from existing literature on face validity within the broader context of simulation modeling and analysis. We plan to conduct a more extensive user evaluation, such as a discussion panel, to assess the practical relevance and generalizability of our findings. Additionally, implementing a more formalized or automated approach could facilitate a more systematic face validity assessment, particularly as our approach primarily outlines what should be done without specifying how it can be achieved (e.g., selecting appropriate process mining techniques). Finally, it would be interesting to apply our proposed method to a simulation model that excels in terms of KPIs but struggles to simulate accurately outlier behavior.

Acknowledgements. We acknowledge the Helmholtz Information & Data Science Academy (HIDA) for providing financial support enabling a short-term research stay at Karlsruhe Institute of Technology (KIT), Germany. We express our gratitude to Ruben Govers for his assistance with the simulation study.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19345-3
2. Alibrahim, A., Wu, S.: Modelling competition in health care markets as a complex adaptive system: an agent-based framework. Health Syst. **9**(3), 212–225 (2020)
3. Balci, O.: Validation, verification, and testing techniques throughout the life cycle of a simulation study. In: Proceedings of Winter Simulation Conference, pp. 215–220. IEEE (1994)
4. Bemthuis, R., Govers, R., Lazarova-Molnar, S.: Using process mining for face validity assessment in agent-based simulation models: an exploratory case study. In: Cooperative Information Systems (in press)
5. Bemthuis, R., Mes, M., Iacob, M.E., Havinga, P.: Using agent-based simulation for emergent behavior detection in cyber-physical systems. In: 2020 Winter Simulation Conference (WSC), pp. 230–241. IEEE (2020)
6. Bemthuis, R.H., Koot, M., Mes, M.R., Bukhsh, F.A., Iacob, M.E., Meratnia, N.: An agent-based process mining architecture for emergent behavior analysis. In: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), pp. 54–64. IEEE (2019)
7. Bemthuis, R.H., Lazarova-Molnar, S.: Discovering agent models using process mining: Initial approach and a case study. In: 2022 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 163–172 (2022)
8. Cenani, S., Arentze, T.A., Timmermans, H.J.P.: Agent-based modeling of cognitive learning of dynamic activity-travel patterns. Transp. Lett. **5**(4), 184–200 (2013)
9. Cooley, P., Solano, E.: Agent-based model (ABM) validation considerations. In: Proceedings of the SIMUL 2011, The Third International Conference on Advances in System Simulation, pp. 134–139 (2011)
10. Day, R.S.: Challenges of biological realism and validation in simulation-based medical education. Artif. Intell. Med. **38**(1), 47–66 (2006)
11. Foramitti, J.: AgentPy: A package for agent-based modeling in Python. J. Open Source Softw. **6**(62), 3065 (2021)
12. Gomes, A.F.D., de Lacerda, A.C.W.G., da Silva Fialho, J.R.: Comparative analysis of process mining algorithms in process discover. In: de Paz Santana, J.F., de la Iglesia, D.H., López Rivero, A.J. (eds.) DiTTEt 2021. AISC, vol. 1410, pp. 258–270. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87687-6_25
13. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75183-0_24
14. Günther, C.W., Rozinat, A.: Disco: discover your processes. BPM (Demos) **940**(1), 40–44 (2012)
15. Klügl, F.: A validation methodology for agent-based simulations. In: Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 39–43 (2008)
16. Law, A.M.: How to build valid and credible simulation models. In: 2019 Winter Simulation Conference (WSC), pp. 1402–1414. IEEE (2019)
17. Leemans, S.J.J., Partington, A., Karnon, J., Wynn, M.T.: Process mining for healthcare decision analytics with micro-costing estimations. Artif. Intell. Med. **135**, 102473 (2023)

18. Llacay, B., Peffer, G.: Using realistic trading strategies in an agent-based stock market model. Comput. Math. Organ. Theory **24**, 308–350 (2018)
19. Martínez-Miranda, J., Pavón, J.: Modeling the influence of trust on work team performance. Simulation **88**(4), 408–436 (2012)
20. Midgley, D., Marks, R., Kunchamwar, D.: Building and assurance of agent-based models: an example and challenge to the field. J. Bus. Res. **60**(8), 884–893 (2007)
21. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manag. Inf. Syst. **24**(3), 45–77 (2007)
22. Robinson, S.: Simulation model verification and validation: increasing the users' confidence. In: Proceedings of the 29th Conference on Winter Simulation, pp. 53–59 (1997)
23. dos Santos Garcia, C., et al.: Process mining techniques and applications-a systematic mapping study. Expert Syst. Appl. **133**, 260–295 (2019)
24. Sargent, R.G.: Validation and verification of simulation models. In: Proceedings of the 24th Conference on Winter Simulation, pp. 104–114 (1992)
25. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2010 Winter Simulation Conference, pp. 166–183. IEEE (2010)
26. Schelling, T.C.: Dynamic models of segregation. J. Math. Sociol. **1**(2), 143–186 (1971)
27. Seid, M., Bridgeland, D., Bridgeland, A., Hartley, D.M.: A collaborative learning health system agent-based model: computational and face validity. Learn. Health Syst. **5**(3), e10261 (2021)
28. Xiang, X., Kennedy, R., Madey, G., Cabaniss, S.: Verification and validation of agent-based scientific simulation models. In: Agent-Directed Simulation Conference, pp. 47–55 (2005)

Progressing from Process Mining Insights to Process Improvement: Challenges and Recommendations

Vinicius Stein Dani[1](✉), Henrik Leopold[2], Jan Martijn E. M. van der Werf[1], and Hajo A. Reijers[1]

[1] Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands
{v.steindani,j.m.e.m.vanderwerf,h.a.reijers}@uu.nl
[2] Kühne Logistics University, Großer Grasbrook 17, 20457 Hamburg, Germany
henrik.leopold@the-klu.org

Abstract. Many organizations have adopted process mining to analyze their business processes, gain insights into their performance, and identify improvement opportunities. Several academic case studies and reports from practice leave no doubt that process mining tools can deliver substantial value to organizations and help them to realize improvements. However, both organizations and academics have also realized that the path from obtaining insights via process mining to realizing the desired improvements is far from trivial. Existing process mining methodologies pay little to no attention to this matter and mainly focus on how to obtain insights through process mining. In this paper, we address this research gap by conducting a qualitative study based on 17 semi-structured interviews. We identify seven challenges pertaining to translating process mining insights into process improvements. Furthermore, we provide five specific recommendations for practitioners and stakeholders that should be considered before starting a new process mining initiative. By doing so, we aim to close the gap between insights and action and help organizations to effectively use process mining to realize process improvements.

Keywords: process mining · insights to action · process improvement · challenges

1 Introduction

Over the last years, many organizations have adopted process mining to analyze their business processes, gain insights into their performance, and identify improvement opportunities [19,33]. Countless academic case studies [33] and reports from practice [19] leave no doubt that process mining tools can deliver substantial value to organizations and help them to realize improvements with respect to relevant performance indicators, such as throughput time [43], conformance [22], or customer satisfaction [14]. However, both organizations and

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 152–168, 2024.
https://doi.org/10.1007/978-3-031-46587-1_9

academics have also realized that the path from obtaining insights via process mining to realizing the desired improvements is far from trivial [16]. In fact, moving beyond diagnostics has been identified as one of the current key challenges of process mining [1]. Existing process mining methodologies, such as Process Diagnostics [9], L* [42], or PM^2 [15], pay little attention to this matter and mainly focus on how to obtain insights through process mining. Some recent papers have contributed to the discourse by investigating how process mining insights can trigger automated actions [6,28,29]. They, however, take a rather technical perspective and do not consider organizational concerns or challenges.

In this paper, we address this research gap and set out to understand the challenges that arise on the path from translating process mining insights into process improvements. Specifically, we aim to answer the following research question: "*Which challenges do organizations face when translating process mining insights into process improvements?*". To answer this question, we conducted a qualitative study based on semi-structured interviews with 17 process mining experts. In this way, we were able to detect seven challenges that organizations have to overcome in this context. Based on the identified challenges, we further derive five specific recommendations that can help organizations making a successful transition from process mining insights to process improvements. With the detected challenges and recommendations, we contribute to the stream of process mining literature that is concerned with process mining methodologies [9,15,42]. Specifically, we extend their scope by providing guidance for the final step in a process mining project.

The rest of the paper is structured as follows. Section 2 introduces the background and the research gap. Section 3 describes our research method. Section 4 presents the identified challenges of translating process mining insights into process improvements. Section 5 reflects on our findings and provides the recommendations we derived. Finally, Sect. 6 concludes the paper.

2 Background

In this section, we discuss the background of our research. Our objective is to demonstrate to what extent existing research focuses on the translation of process mining insights into process improvements. To this end, we first review existing process mining methodologies. Then, we reflect on how process mining insights have been used across different studies.

2.1 Process Mining Methodologies

The effective use of process mining for process improvement is often a complex endeavor that goes way beyond the use of process mining software [16]. Process mining methodologies, therefore, aim to provide a reference structure for the application of process mining by defining a number of specific steps. Among others, those steps include scope definition, data collection, the application of process mining techniques such as discovery and conformance checking, result analysis, and process improvement [16]. Several such process mining methodologies have been defined in the past, the most prominent being the process diagnostics methodology (PDM) [9], L* [42], and PM^2 [15].

While these methodologies differ with respect to several details, they have two main things in common. First, they propose a similar high-level flow involving steps such as data collection, application of process mining techniques, and result analysis. Second, they only pay little attention to how process mining insights can be translated into process improvements. At the same time, however, they acknowledge that this step is important. The authors of PDM highlight that the interpretation of the insights identified through their methodology is critical but lies in the responsibility of the organization [9]. The authors of L* explain that their methodology can lead to four different improvement actions: redesigning, adjusting, intervening, and supporting. Yet, they only discuss a few examples of what each action entails and do not reflect on how those actions can be implemented [42]. Also, the authors of PM^2 explicitly acknowledge the importance of process improvement based on the obtained insights by including a step called *process improvement and support*. They, however, argue that the realization of such improvements is typically done in the context of a separate project [15].

This lack of attention with respect to the translation of insights into improvements is also discussed in a relatively recent meta study of process mining case studies by Emamjome et al. [16]. They point out that the last phase of process mining projects is only superficially considered in the analyzed studies and, hence, has a low degree of "thoroughness". They conclude that most case studies they analyzed fit somewhere between the following two categories: 1) "the studies provide insights without any recommendation", and 2) "the studies provide some recommendations on how to improve the process(es), but do not refer to any implementation".

To understand how process mining insights are actually used in real-life cases, we review respective literature in the next section.

2.2 Use of Process Mining Insights

The value an organization can realize through process mining highly depends on what the organization does with the obtained insights. Recognizing this, many researchers investigated how process mining insights are used or can be used. In general, we can distinguish three main categories for the use of process mining insights: 1) supporting process understanding and documentation, 2) improving the investigated process, and 3) improving information system(s) supporting the investigated process. Below, we briefly elaborate on each category.

Using process mining insights to support *process understanding and documentation* relates to the explorative use of process mining. Simply put, process mining can help organizations to understand what is going on inside their organization. Besides the discovery of the control flow [5], i.e., the order of activities, process mining can also help to uncover how resources interact [4] or to identify business rules [10,22]. Some authors highlight the importance of writing [30] and presenting [31] reports based on the acquired process mining insights, yielding documentation creation, reviewing, or updating. For a more comprehensive overview, we refer the interested reader to the literature study from Garcia et al. [37].

In line with the main objective of process mining, many authors aim to use process mining insights to *improve the investigated process* by generating respective recommendations. Such recommendations can be generic and refer to process change or, simply, redesign [18,27]. Some, however, are more specific and include preventing a specific activity from happening [14], eliminating an activity [3], or increasing the frequency of a specific activity [10]. Works focusing on the resource perspective suggest actions such as adding resources [12,13] or increasing resource involvement [38].

As the execution of many processes is supported by one or more information systems, process mining insights can also reveal how to *improve those information systems* in different ways. Some authors discuss rather general aspects such as improving the information system's usability [23,35,40]. Other studies report on redefining [39] and adjusting [25] specific feature settings to be more permissive or restrictive based on thresholds identified through process mining. There are also studies reporting on testing new information system features [21] or identifying opportunities for implementing automation [17,32].

The brief review above illustrates that process mining insights can provide valuable input for both understanding and improving processes and the associated information systems. However, what is currently missing is a clear path towards implementation. As an example, consider a scenario where process mining insights are used to recommend the introduction of an additional quality check in a process. While this recommendation is useful, especially because it is based on a data-driven analysis of the underlying process, putting this recommendation into action is far from trivial. Among others, this requires commitment from both the process manager and the process participants, proper communication of the changes, an allocation of the required resources, additional training, etc. While several authors discuss the importance of communication [2,26,44] and also training in such contexts [43], these concerns are generally only superficially considered. As a result, it remains unclear which challenges need to be overcome to translate insights (or recommendations based on insights) into process improvements.

With this paper, we aim to close this gap by identifying and understanding the challenges that occur in this context. In the next section, we explain the methodology of our study.

3 Research Method

To identify and understand the challenges that need to be overcome to translate process mining insights into process improvements, we interviewed 17 experts with several years of industrial experience in process mining projects. Below, we describe our research method. Specifically, we elaborate on the definition of the target population and the interview protocol, the data collection, and the data analysis.

3.1 Definition of Target Population and Interview Protocol

Driven by our research question, our target population included process analysts, business analysts, and researchers with experience in process mining projects in industry. We defined our semi-structured interview protocol consisting of a set of predefined open-ended questions inspired by [8,11]. The intention was to understand the interviewees' experiences and perspectives related to what happens with process mining insights after they have been acquired. We conducted a test run of our interview protocol with two participants that are not part of this research. With this test run, we verified that the predefined questions were well suited to obtain the desired insights.

3.2 Data Collection

We sent personal invitations to potential participants via e-mail and LinkedIn. In total, we interviewed 17 process mining experts. Table 1 provides an overview of the interviewees. It shows the interviewees' job title (where PM stands for *Process Mining*), experience with process mining in years (cf. column *Exp.*), as well as the continent, size and domain (where IT stands for *Information Technology*) of the organization they work for. The interviewees have an average

Table 1. Interviewees' demographics

Ref.	Job title	Exp.	Organization		
			Continent	Size	Domain
I1	Business Analyst	5–10	Asia	201–500	Oil and Gas
I2	PM Consultant	10–15	Europe	51–200	IT
I3	Transformation Consultant	5–10	Europe	1k–5k	IT
I4	PM Consultant	5–10	Europe	1k–5k	IT
I5	PM Product Owner	10–15	Europe	1k–5k	Finance
I6	Researcher/PM Consultant	10–15	Europe	1k–5k	Education
I7	Researcher/PM Consultant	15–20	Europe	1k–5k	Education
I8	PM Specialist	5–10	Europe	>10k	Public
I9	Senior Manager	10–15	Europe	>10k	Finance
I10	PM Specialist	10–15	Europe	>10k	Audit
I11	PM Specialist	10–15	Europe	>10k	Public
I12	PM Product Owner	15–20	Europe	>10k	Healthcare
I13	PM Product Owner	5–10	North America	1k–5k	IT
I14	PM Consultant	5–10	North America	>10k	IT
I15	PM Analyst	1–5	Oceania	1k–5k	Food
I16	Researcher/PM Consultant	10–15	Oceania	1k–5k	Education
I17	PM Product Owner	5–10	South America	51–200	IT

of seven years of industrial experience with process mining. Eleven of them also obtained a PhD in the process mining field and, therefore, also had additional exposure to the subject. The interviewees used a large variety of process mining tools including ARIS Process Mining, Celonis, Fluxicon Disco, Minit, PAFnow, ProM, UpFlux, UiPath Process Mining, and SAP Signavio.

The interviews were conducted as follows. First, we asked the participants a couple of questions about themselves such as "*What is your role with respect to process mining in your organization?*" and "*For how long have you been working with process mining?*". Next, we asked general questions about the process mining projects, such as "*What usually triggers the use of process mining in your organization?*" and "*What are usually the expected insights from the stakeholders of a process mining initiative?*". Then, we asked them to share some details about process mining projects they have been involved with and to talk about the process that was under investigation, the effort that was required to acquire the insights, and which main insights were obtained. Finally, we asked: "*What happened to the process mining insights after they have been acquired?*". On average, the interviews lasted 54 min.

3.3 Data Analysis

Each interview was audio-recorded and transcribed. Then, we anonymized the transcriptions by removing any information that could reveal the interviewees' identity or the organization they worked for. We conducted a qualitative coding using four main steps [11,36]. First, we familiarized ourselves with the interviews by reading them and taking general notes. Second, we re-read the interviews and wrote memos. For example, when an interviewee talked about the customer expectations being much different from the process mining outcomes such that they decided to discontinue the project, we added memo notes such as "expectation" and "project ends". Third, we reviewed our codes to identify possible connections among the codes or the possibility of merging multiple codes into higher-level categories. Finally, we identified multiple categories concerning challenges relating to translating process mining insights into process improvements.

4 Findings

In this section, we present the findings of our study. In total, we identified seven specific challenges that can impair an organization's ability to translate process mining insights into process improvements. We classified these seven challenges into three main categories: 1) organizational commitment, 2) expertise, and 3) expectations. In the subsequent sections, we elaborate on each category in detail and illustrate the respective challenges by using quotes from our interviews. An overview of the three main categories and the seven challenges, as well as the number of supporting interviewees for each category, is depicted in Fig. 1.

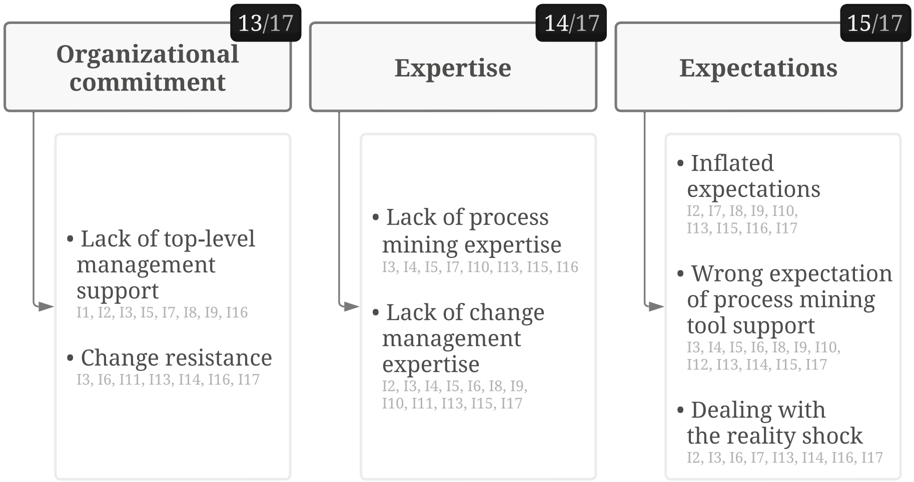


Fig. 1. Overview of the identified challenges

4.1 Organizational Commitment

The fact that change requires organizational commitment has been emphasized in BPM literature for a long time [7,20,41]. However, our interviews revealed that this awareness is often limited when it comes to the application of process mining. We identified two specific challenges in this context: lack of top-level management support and change resistance.

Lack of Top-Level Management Support. Several interviewees highlighted that, without support from top-level management, process mining projects do not yield much besides acquiring insights about the analyzed processes. For instance, Interviewee I2 pointed out that "*in order to have changes, you do need some top-level support because you need a budget. Another reason why you need top-level support is for them to be able to say that now it's part of the vision and we should spend time with it, it is part of the initiatives to actually improve this*". This point is also supported by interviewee I16, who reported on a project where process mining insights were successfully translated into process improvements: "*(...) but you need to understand that this is the head of the organization. He wants to improve the process. If he wasn't the CEO, if he was a developer, for example, I do not think he would be able to make this change.*"

Interviewee I3 also emphasized the importance of involving a manager or director (i.e., non c-level managers) who can understand and deliver the insights to their team because "*then you get into the normal psychology of human change by having an ambassador, a leader who says we need to change, and supports the work required to change*". Similarly, I8 stated that "*the organization started to check and act upon process mining insights because there was a new program manager that really believed in process mining*".

Change Resistance. Resistance to change is a well-known and well-studied phenomenon [34]. In our interviews, we encountered both individual as well as organizational resistance to change. As for individual resistance, the interviewees mentioned instances of resistance that can be related to *habit* and *fear of the unknown. Habit* describes the problem of people resisting change because they need to alter the way they work. Resistance due to a *fear of the unknown* is more abstract and can be attributed to the uncertainty that individuals experience when changes are introduced. As for organizational resistance, interviewees mentioned instances of *structural inertia* and *threat to established power relationships. Structural inertia* refers to changes that interfere with the organizations' mechanisms built to produce stability in the work processes. The *threat to established power relationships* occurs when these power relationships are at risk because of the redistribution of the responsibility for decision-making.

Interviewee I6 described a case that relates both to *habit* and *structural inertia*: "*... it is important to grow confidence on process mining with smaller suggestions for improvement first, and really think through which kind of recommendations of improvements to make, because asking someone to change the way that he or she works might not be the smartest way of going for it*". Interviewee I6 also shared an interesting reflection on how they handled the anticipated resistance to change: "*the stakeholders already know that I am not going to try to replace anyone or any decision, I will try only to support or try to provide information or ways for them to do their work just as they were doing before but with very small changes. Just then is when people start accepting the suggestions to change*". This example highlights how important it is to involve someone in which the affected people in the organization trust. In this particular case, the interviewee has been responsible for different process mining projects in the organization for almost four years.

Interviewees I13 and I14 also described examples of *fear of the unknown.* I13 mentioned that "*... some people get 'cold feet' about going forward with process mining projects because people will demand responses from them later and they are just afraid to take the responsibility*". Similarly, I14 pointed out that "*people are too scared of having to change*". They both also mentioned cases of people impeding the process mining project by hiding information. Interviewee I13 stated that "*sometimes we could show people up in the value chain, directors, the potential value of process mining, and they would sponsor our conversations with the operations team who, however, didn't want their directors to know everything that was going on within operations. They didn't want to be monitored. Then, they told their directors that they did not have all the data or did not find anything meaningful*". Interviewee I14 also shared a case where information was hidden by managers, stating that "*managers know they can do better, but they also know that their bosses do not know they can do better. So they can play 'life easy', and using process mining would take this advantage away from them*".

4.2 Expertise

Realizing process improvements through process mining requires the organization to have certain expertise at its disposal. On the one hand, it is critical that the process mining insights can be properly understood and interpreted, i.e., there is a need for process mining expertise. On the other hand, identified weaknesses must also lead to effective changes in the organization's processes, i.e., there is also an immanent need for change management expertise.

Lack of Process Mining Expertise. Several interviewees pointed out that, according to their experience, the output of process mining tools can hardly be properly interpreted without an employee who is capable of understanding both process mining as well as the domain. Interviewee I4 mentioned that in one of the organizations they worked on, the organization had "*purchased the license, and they were supposed to use the tool themselves, but they weren't able to generate any findings or insights*". While the interviewee, having several years of process mining experience, was not specifically hired for that project, they had to step in to prevent the process mining project from being canceled.

Interviewees I5 referred to "*the need for a process mining expert working in the project, especially one that can also learn or previously know about the domain*". Similarly, interviewee I16 stated that "*process mining is a good tool for communication within the team if they are interested from the beginning, but process mining needs a good process analyst and involvement of a domain expert*". Also, interviewee I7 mentioned that "*you need a process mining expert who can translate the event log data into insights to the organization*", and interviewee I13 mentioned that the "*big blockers to buying and using process mining are that companies over and over again say that they do not have the people to analyze what process mining is showing them*".

Lack of Change Management Expertise. Several interviewees highlighted that there are different cases that demand for an (impartial) change management expert. In some cases, there is a lack of technical expertise and no commitment from the stakeholders to work on the changes. For example, interviewee I4 mentioned that the proposed changes based on process mining insights were never implemented because "*it tends to be complicated making changes and running two configurations in the same live data. We did not have the expertise to make these changes, and the stakeholders didn't want to commit with their own resources to do it*".

In other cases, as reported by interviewee I11, there is a lack of financial support and of a manager with company-wide access. Interviewee I11 mentioned that the impact they could make was "*initially small to nonexistent, because they needed a strong manager to bring widespread process mining initiatives in the organization, make these initiatives continuous and more effective, but this manager was not there*". Interviewee I11 also mentioned that when there was a "*strong manager*", related to the financial department of the company, he was capable of implementing a company-wise widespread process mining initiative. According to the interviewee, this manager had broad access to different departments and financial support.

Finally, as raised by interviewee I17, there should be an impartial change manager to deal with political-related aspects that are harming the company: "*... [we] should put someone capable to do the change management in the company to recover the lost money caused by inefficient employees*". This, however, did not happen because, in this case, "*a very well related person, that does not follow good practices can be protected by their peers. We can detect such behaviors, but nothing happens, and the project ends*".

4.3 Expectations

We found that the application of process mining is often associated with high expectations and partially also with misconceptions. We identified three main challenges in this context: inflated expectations, wrong expectation of process mining tool support, and dealing with the reality shock.

Inflated Expectations. Several interviewees highlighted the importance of being aware of the effort required to translate process mining insights into improvements, and not expect that process mining will magically improve the process.

Interviewee I13 shared that "*people have been disappointed with process mining in the past, but mostly because either they had inflated expectations or they underestimated the work that needs to go into turning insights into something useful*". According to interviewee I17, "*the most successful projects have a good alignment between expectations and insights*". The interviewee mentioned that they drive this alignment based on previous experience and previously defined templates building on expected and acquired insights.

Interviewee I2 suggested to handle inflated expectations by starting the process mining project small: "*oftentimes it is difficult to turn process mining insights into value, because if you find something, then you might not know, for example, the person whose responsibility this is to pick it up. It also might be a not known pain point, which would lead you to first needing to convince people that actually what you found is true. So, my approach is to typically start small, not with the biggest money maker process, to start gaining some trust in the solution and start with problems that people already know and about which they might already have some hypothesis*". According to interviewee I2, starting small also makes it easier for the company to acquire experience using process mining and understand how fast the company is in implementing, analyzing, and getting value out of the process mining project. Similarly, to narrow down stakeholder's expectations, interviewee I9 mentioned that "*before starting any project we always sit together with the client, ask them about their priorities, and also whether there is any specific challenge that they would like us to focus on, or that they would like more insights about or more recommendations about*".

Wrong Expectation of Process Mining Tool Support. This challenge relates to the problem that stakeholders still see process mining as a "*full-fledged process improver setup*", which is not the case. Therefore, process mining methodologies and advocates should consider including change management initiatives as one of its stages, or at least provide initial guidance regarding the effort required to move process mining insights into action.

Interviewee I4 mentioned that "*process mining should not be the only tool or artifact for process improvement. It should be aligned with other tools and initiatives for that. Data itself is not enough to really understand the underlying problem. With data and process mining we can, most of the time, describe the problem well, but we can't really say how to improve it; there needs to be some sort of process understanding that then is used to finally improve the process*". Interviewee I12 highlighted that "*process mining is a tool to support process redesign initiatives in the organization. So, an advice for making process mining more usable, the process should be analyzed and then there should be a second phase to work upon improvements based on what we saw*".

Also interviewee I15 highlighted that "*the limitation of the tool compared to the expectations of the stakeholder is a challenge. Process mining requires a few stages to actually bring value to the customers: we need to build the data model, then do the analysis, then, based on the insights, think of how to turn insights into action. And the action part is the challenging part for business. Turning insights into action is certainly a pain point for most businesses. Turning insights into action involves different departments; it involves how the business operated before and how they are going to operate in the future, and the most challenging part is that it involves multiple departments, and it really depends on how the senior managers are going to do. It really depends on how you manage your company*".

Interviewee I13 shared a situation that they went through when after they showed their process mining tool to a friend that was working at a big tech company, this friend asked them: "*are you telling me that I should pay you money for you to show me my problems and not solve them? Really?*". According to I13, "*there is a need for expectation management and, of course, this inflated-expectations is not a problem exclusive to process mining. And as long as process mining is not something that is well understood by the market, inflated expectations will always be there*".

Dealing with the Reality Shock. Process mining is a "*big mouth*" and it will uncover "*hard truths to swallow*". While some interviewees mention that "*it is easier for managers trying to use process mining to say that it doesn't work than to accept the insights it can deliver*" (I13) and "*process mining is too truthful*" (I14), interviewee I3 suggests a mean to deal with the reality shock: "*you need to involve your customer because then they evolve in the way of thinking at the same rate as you. If you don't do that and you simply take the data, go back to your cave and start analyzing it, you come back conceptually and mentally three steps ahead of them, and if you then just drop it on them, they could be very defensive because you basically tell them their process is a mess, and that's very often what it is. So, you need to take them along on the journey.*".

The reality shock can occur for the organization conducting a process mining project to understand and improve their own processes, for the process participants, and it can also be for service providers or process analysts. An example of a reality shock for the organization, interviewee I6 shared a case where the nurses of a hospital were highly stressed with their work. The managers of the hospital did not understand how that could be, considering how much idle time the nurses had, based on usual process discovery-related insights acquired. The interviewee decided to look more closely at the daily work of the nurses by conducting observation sessions. They learned that the idle time was just a reflection of limited data availability related to their daily work process. The managers learned the hard way that process mining can only show what is included in the event log. All the times, the nurses needed to hurry to a patient's room to attend to a patient's call had not been recorded in any information system.

An example of a reality shock for the process participant, also shared by interviewee I6, related to long waiting times for an emergency room. At first, the physicians were not enthusiastic about the process mining project that was started by the management team. The interviewee learned that one of the reasons for this long wait was that physicians switch context too often. In other words, the doctor has a certain amount of patients in the waiting room; one has an orthopedic problem, the other one has a cardiac problem, the other a neurological problem, etc. The physicians did not notice, but they were taking too long to think of the different special reasons related to different patient needs. The interviewee suggested them to group patients per type of complaint and analyze each group together. They applied the suggestion to one department and could see the waiting time of all patients reduced by 20%. Thereafter, the physicians started to accept the technique.

As an example of a reality shock for the process mining service provider, interviewee I13 shared an example related to a credit card sales process. In essence, the process was concerned with selling a credit card to clients in a physical store. Part of the selling process was a credit analysis to check the customer's credit status. The analysis revealed that every time a human was involved in the credit analysis, it took double the time to close the sale, and the likelihood of a successful sale decreased. After analyzing the credit analyst's actual work, the interviewee noticed they were very fast. They further investigated this inefficiency and learned that whenever a manual credit analysis was triggered, the client in the store was said to wait and would walk around the store and eventually simply leave. The interviewee suggested a very simple solution to this problem (e.g., offer coffee to the client or talk to them for a while), but "*once the manager of the credit checking group realized that it wasn't the credit analysis that was delaying the process, it was not his fault and he didn't care about making any changes anymore and they did not continue using process mining after that*".

5 Recommendations

The findings from our interviews reveal that translating process mining insights into process improvements comes with substantial challenges. Our interviews also highlight that it is likely that the transition from insights to improvements is never made if these challenges remain unaddressed. It is not particularly surprising that several of the challenges we identified relate to phenomena that have been made in the context of change management, such as resistance to change [24]. Yet, process mining projects, and hence also the associated challenges, differ from traditional change management projects, digital transformation projects, and process redesign initiatives. Most importantly, in process mining projects, the insights that provide the starting point and argument for changes are acquired through software. Naturally, this does not only changes the nature of change resistance but also calls for specific expertise for interpreting results and implementing changes. As existing process mining methodologies have paid little to no attention to these aspects [16], we derived five recommendations that organizations should consider when starting a process mining initiative. The recommendations provide specific input on how process mining projects should be prepared, set up, conducted and who should be involved. Specifically, our derived recommendations for process mining projects in practice are the following:

R1 - Engage top-level management support: Top-level management support should be secured before the start of the process mining initiative. It is essential for getting appropriate financial support, conveying the importance of the initiative, and ensuring the ability to actually implement the required changes.

R2 - Be ready to face resistance to change: Resistance to change must be expected in every process mining initiative and should be handled appropriately. We found that it is particularly about communication. If people understand which changes will be implemented and why, they are much more likely to support their implementation. Handling fears and concerns, therefore, is a critical activity.

R3 - Have process mining and domain expertise at your disposal: One of the critical steps in every process mining initiative is the interpretation of the acquired results. This requires an individual who is familiar with both process mining and the respective domain of the organization. Such a person should be either hired or educated on time.

R4 - Have change management competence at your disposal: Translating process mining insights into process improvements requires change. Hence, it is essential to have change management expertise available in the organization. Such a change manager will follow up on the recommendations of the process analyst (see R3) and develop a strategy on how to successfully implement the desired changes.

R5 - Manage expectations: Expectations among several stakeholders of a process mining initiative are often unrealistic. Therefore, it is important to manage expectations with respect to the outcome and also the effort that will be required to realize process improvements through process mining. People need to be aware that process mining is a tool and will not magically improve processes without any effort.

The recommendations stem from the identified challenges (cf., Sect. 4). For convenience, Table 2 presents which recommendations address which challenges.

Table 2. Recommendations to challenges mapping

R1	R2	R3	R4	R5	Challenge
•					Lack of top-level management support
	•				Change resistance
		•			Lack of process mining expertise
			•		Lack of change management expertise
				•	Inflated expectations
				•	Wrong expectation of process mining tool support
				•	Dealing with the reality shock

6 Conclusion

In this paper, we investigated which challenges organizations face when translating process mining insights into process improvements. To this end, we conducted a qualitative study involving 17 interviews with process mining experts. Based on these interviews, we identified seven challenges, which we turned into five specific recommendations that organizations using process mining should consider. Among others, we highlighted the importance of top-level management support and the availability of expertise with respect to process mining, the domain, and change management. After all, turning process mining insights into improvement requires change and, therefore, also a respective commitment from several levels of the organization.

Naturally, our study is subject to limitations. Most importantly, our study is qualitative and, hence, limited in terms of generalizability. We, however, attempted to mitigate this concern by involving process mining experts that worked in different organizations and settings, have used different process mining tools and approaches, and faced different problems in their organizations. Other biases, e.g. with respect to the analysis, we mitigated by jointly building the data collection protocol, and jointly conducting, reviewing, and discussing the coding effort related to the data analysis. Therefore, we are confident that our results appropriately reflect the challenges organizations face, and provide valuable input about how process mining insights can be translated into process improvements.

In future work, we aim to validate our findings in the context of a large case study. Furthermore, we plan to incorporate our findings into a comprehensive proposal for a process mining methodology.

Acknowledgements. Part of this research was funded by NWO (Netherlands Organisation for Scientific Research) project number 16672.

References

1. van der Aalst, W.M.P., Carmona, J.: Scaling process mining to turn insights into actions. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 495–502. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_17
2. van der Aalst, W., et al.: Business process mining: an industrial application. Inf. Syst. **32**, 713–732 (2007)
3. Aksu, U., Reijers, H.A.: How business process benchmarks enable organizations to improve performance. In: EDOC, IEEE (2020)
4. Alvarez, C., et al.: Discovering role interaction models in the emergency room using process mining. J. Biomed. Inf. **78**, 60–77 (2018)
5. Augusto, A., et al.: Automated discovery of process models from event logs: review and benchmark. TKDE **31**, 686–705 (2018)
6. Badakhshan, P., Bernhart, G., Geyer-Klingeberg, J., Nakladal, J., Schenk, S., Vogelgesang, T.: The action engine - turning process insights into action. In: Demo Track at ICPM (2019)
7. Bandara, W., Indulska, M., Chong, S., Sadiq, S.: Major issues in business process management: an expert perspective. In: ECIS, pp. 1240–1251 (2007)
8. Boyce, C., Neale, P., International, P.: Conducting in-depth interviews: a guide for designing and conducting in-depth interviews for evaluation input. In: PI (2006)
9. Bozkaya, M., Gabriels, J., van der Werf, J.M.: Process diagnostics: a method based on process mining. In: eKNOW, pp. 22–27 (2009)
10. Bozorgi, Z.D., Teinemaa, I., Dumas, M., Rosa, M.L., Polyvyanyy, A.: Process mining meets causal machine learning: discovering causal rules from event logs. In: ICPM. IEEE (2020)
11. Bryman, A.: Social Research Methods. Oxford University Press, Oxford (2016)
12. Cela, O., Front, A., Rieu, D.: CEFOP: a method for the continual evolution of organisational processes. In: RCIS, IEEE (2017)
13. Cho, M., Song, M., Comuzzi, M., Yoo, S.: Evaluating the effect of best practices for business process redesign: an evidence-based approach based on process mining techniques. Decis. Support Syst. **104**, 92–103 (2017)
14. Dees, M., de Leoni, M., van der Aalst, W., Reijers, H.: What if process predictions are not followed by good recommendations? In: BPM Ind. Forum, pp. 61–72 (2019)
15. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM^2: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19
16. Emamjome, F., Andrews, R., ter Hofstede, A.H.M.: A case study lens on process mining in practice. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 127–145. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_8
17. Esiefarienrhe, B.M., Omolewa, I.D.: Application of process mining to medical billing using L∗ life cycle model. In: ICECET, IEEE (2021)
18. Goel, K., Leemans, S.J.J., Wynn, M.T., ter Hofstede, A.H.M., Barnes, J.: Improving PhD student journeys: insights from an Australian higher education institution. In: BPM Ind. Forum, pp. 27–38. CEUR-WS.org (2021)
19. Grisold, T., Mendling, J., Otto, M., Brocke, J.V.: Adoption, use and management of process mining in practice. Bus. Process Manag. J. (2020)

20. Hammer, M., Champy, J.: Reengineering the corporation: a manifesto for business revolution. HarperCollins, Collins Bus. Essent. (2009)
21. Huang, C., Cai, H., Li, Y., Du, J., Bu, F., Jiang, L.: A process mining based service composition approach for mobile information systems. In: MIS, pp. 1–13 (2017)
22. Jans, M., Hosseinpour, M.: How active learning and process mining can act as continuous auditing catalyst. IJAIS **32**, 44–58 (2019)
23. Kudo, M., Nogayama, T., Ishida, A., Abe, M.: Business process analysis and real-world application scenarios. In: SITIS, IEEE (2013)
24. Lauer, T.: Success Factor Person: Right Leadership in Change. In: Change Management, pp. 83–106. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-662-62187-5_6
25. Lee, C., Ho, G., Choy, K., Pang, G.: A RFID-based recursive process mining system for quality assurance in the garment industry. In: IJPR, pp. 4216–4238 (2013)
26. Leemans, M., van der Aalst, W.M.P., van den Brand, M.G.J., Schiffelers, R.R.H., Lensink, L.: Software process analysis methodology – a methodology based on lessons learned in embracing legacy software. In: ICSME, IEEE (2018)
27. Mahendrawathi, E., Zayin, S.O., Pamungkas, F.J.: ERP post implementation review with process mining: a case of procurement process. PCS **124**, 216–223 (2017)
28. Park, G., Aalst, W.: Action-oriented process mining: bridging the gap between insights and actions. Progress Artif. Intell. 1–22 (2022). https://doi.org/10.1007/s13748-022-00281-7
29. Park, G., van der Aalst, W.M.P.: A general framework for action-oriented process mining. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 206–218. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_16
30. Partington, A., Wynn, M., Suriadi, S., Ouyang, C., Karnon, J.: Process mining for clinical processes. In: TMIS, pp. 1–18 (2015)
31. Polyvyanyy, A., Pika, A., Wynn, M.T., ter Hofstede, A.H.: A systematic approach for discovering causal dependencies between observations and incidents in the health and safety domain. Saf. Sci. **18**, 345–354 (2019)
32. Ramires, F., Sampaio, P.: Process mining and lean six sigma: a novel approach to analyze the supply chain quality of a hospital. In: IJLSS, pp. 594–621 (2021)
33. Reinkemeyer, L.: Process mining in action. In: Process mining in action: principles, use Cases and outlook (2020)
34. Robbins, S., Judge, T.: Organizational Behavior. Pearson Edu. (2016)
35. Rubin, V.A., Mitsyuk, A.A., Lomazova, I.A., van der Aalst, W.M.P.: Process mining can be applied to software too! In: ESEM, ACM (2014)
36. Saldana, J.: The Coding Manual for Qualitative Researchers. SAGE, Los Angeles (2015)
37. dos Santos Garcia, C., et al.: Process mining techniques and applications - a systematic mapping study. In: ESA, pp. 260–295 (2019)
38. Stuit, M., Wortmann, H.: Discovery and analysis of e-mail-driven business processes. Inf. Syst. **37**, 142–168 (2012)
39. Tawakkal, I., Kurniati, A.P., Wisudiawan, G.A.A.: Implementing heuristic miner for information system audit based on DSS01 COBIT5. In: IC3INA, IEEE (2016)
40. Trinkenreich, B., Santos, G., Confort, V., Santoro, F.: Toward using business process intelligence to support incident management metrics selection and service improvement. In: SEKE, KSI (2015)
41. Trkman, P.: The critical success factors of business process management. Int. J. Inf. Manage. **30**, 125–134 (2010)

42. van der Aalst, W.: Process mining: discovery, conformance and enhancement of business processes. Springer (2011)
43. Weerdt, J.D., Schupp, A., Vanderloock, A., Baesens, B.: Process mining for the multi-faceted analysis of business processes - a case study in a financial services organization. Comput. Ind. **64**, 57–67 (2013)
44. Zerbino, P., Aloini, D., Dulmin, R., Mininno, V.: Towards analytics-enabled efficiency improvements in maritime transportation: a case study in a mediterranean port. Sustainability **11**, 4473 (2019)

Developing Taxonomies for Business Process Engineering

Ton Soetekouw[1], Paul Grefen[1,2](✉), Irene Vanderfeesten[3], and Oktay Turetken[1]

[1] Eindhoven University of Technology, Eindhoven, Netherlands
p.w.p.j.grefen@tue.nl
[2] Eviden Digital Transformation Consulting, Amstelveen, Netherlands
[3] KU Leuven, Leuven, Belgium

Abstract. In many business environments, we find extensive business process structures that consist of many individual processes, each with a complex composition of activities. The elements in the processes are often based on an ad-hoc, existing way of working, which is not always properly documented. The processes evolve over time, not rarely on a per-process basis. Consequently, process definitions diverge and the use of process elements within and between process definitions becomes misaligned. To address this issue, we propose the use of catalogs of standardized process building blocks in business process engineering. Different from approaches using patterns, we base our catalogs on foundational parts (which we call *primitives*) organized in three dimensions: business process activities, objects manipulated by activities, and actors performing activities – starting with the semantics of processes, not the syntax. To provide a solid basis for the structuring of each of the dimensions (and hence the organization of the foundational parts in the catalogs), we use taxonomies. In this paper, we discuss the development of these taxonomies. We apply a slightly modified existing taxonomy development method, which uses both deductive and inductive steps. We discuss the development of one taxonomy in detail, basing the inductive steps on processes from a complex, real-world case organization. In doing so, we make a first step towards a business process engineering approach that is centered on a process-content-first point of view, aligned with the needs of the process management practice.

Keywords: Business Process Engineering · Taxonomy · Business Process Element · Business Process Catalog

1 Introduction

The introduction of new technologies and rapid changes in market conditions increase the performance demands put on organizations. Business processes in these organizations control the deployment of their capabilities, both internally and in inter-organizational collaborations. Organizing the business processes into a business process architecture provides the organizations with the capacity to deal with change in a structured way. We

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 169–186, 2024.
https://doi.org/10.1007/978-3-031-46587-1_10

refer to this capacity as agility [24]. The structure of these process architectures has been covered by research, both in the holistic architectural sense [10] and in the ingredients to build the structures in these architectures, mostly in the form of process patterns [1]. The majority of the research efforts in this domain focuses, however, on what we call the *syntax* of process structures, focusing on control flow (or data flow) structures in which process activities are black boxes. Process architectures can also be found as aspects or viewpoints to define behavior in enterprise architecting approaches [27]. Here, process activities are linked, for example, to business services. Like in the process architecture and patterns domains, research in this domain typically focuses on a syntactical point of view. There is not much research yet, however, in addressing the functional content of the business processes, what we call the *semantics* of the activities in the processes. This is where our research aims to contribute, providing a semantics-driven structure to complement the more syntax-driven structure.

To construct the conceptual foundation for our contribution to business process architecture, we undertake a quest for the fundamental elements of operational and informational functionality of business processes [26]. We combine this with the development of an engineering approach that supports the construction of aggregations of functionality from these fundamental elements. The concept of modularity [15] is central to our approach, as modularity enables agility. The quest for the discovery of a foundational, semantics-based modularity concept provides structured access to what we call *business process DNA*. From the foundational elements, standardized as parts, it becomes possible to argue about the application of new ways to construct business processes and the role of parts and behavioral patterns, as is done in [8] and [3].

Using the concept of standardized parts and engineering patterns, resulting from a modular business process architecture, offers the foundation for agile mass customization [12] in business process design, using a proven method to deal with variety. In this endeavor, we investigate the possibility of integrating requirements from the business operations, information systems and information technology domains at the lowest level of functional specification. In our analysis of business process DNA, we take as a starting point that all organizational activity happens through processes [6], the elements of which ultimately boil down to 'actors undertaking activities on objects'. These three foundational elements of functionality structure, properly identified and configured in business process activity blueprints, enable the specification of all constructs of functionality at the lowest level, constituting what we call *primitives* in previous work [26].

In this paper, we take our research further by developing a taxonomy approach as a means for the definition of business process elements. We argue about the applicability of a taxonomy development method [16, 19] in the context of a complex real-world pilot case. This pilot case study is undertaken to identify and shape those elements of business processes that represent distinct properties at the lowest level of specification, meaning that they cannot be further de-factorized without losing their functional meaning. Once we were able to identify these foundational elements, we applied a taxonomy definition method to be able to identify its 'genome'. The resulting taxonomy opens up the dimensions of a catalog of reusable foundational business process-building elements (primitives) and constructs thereof. Working in this way, we aim to develop a method as a DSR artifact that enables rigor in business process construction combined with adaptability in its application, hence providing an adequate basis for true business agility. In this context, the leading research question for this paper is: how can a taxonomy be developed for identifying and classifying elements for a catalog of process primitives?

The pilot case is conducted at Rijkswaterstaat (RWS). RWS is part of the Dutch Ministry of Infrastructure and Water Management and is responsible for the design, construction, management, and maintenance of the main infrastructure facilities in the Netherlands, such as roads, waterways, bridges and tunnels. Thereby, RWS is a key player in the national infrastructure that serves both society and business in the Netherlands. RWS has about 10,000 employees, distributed across a complex and distributed organizational structure. The case study is located in the department at the headquarters of RWS in Utrecht that is responsible for the business processes around the standardization of the design, construction and exploitation of bridges and tunnels across the Netherlands. The department is subdivided into several smaller units. The department as a whole manages a large set of complex business processes. One of the problems that the department faces is that its processes have evolved over many years and now have an alignment issue, making it hard to recognize commonalities (or even overlaps in structure), identify unnecessary differences, and address inefficiencies. In other words, they face the problems of *expansive BPM* and *lack of objectivity in process descriptions* [4]. This makes them interested in participating in our research.

The structure of this paper is as follows. In Sect. 2, we place our work in the perspective of related research. In Sect. 3, we present the methodology used in our work. Section 4 presents the overall process of developing the taxonomy. In Sect. 5, we present in detail the iterative process of building the taxonomy. As this is a novel line of research that requires next steps, we explicitly reflect on the current step in Sect. 6. Section 7 presents conclusions and outlines future work.

2 Related Work

In this section, we present the core of work that is related to our research. We do this from two perspectives. The first perspective is that of other approaches towards the identification of business process building blocks, i.e., focusing on the 'product' perspective of our work. As explained in the introduction, we do this from the process activity content point of view, not from the control flow (or data flow) point of view typically found in pattern-oriented approaches. Our point of view is related to what is mentioned as the concept of 'content patterns', used as one of 9 categories in a 'process pattern taxonomy' [13] as a classification of process building block classifications. The second perspective of related work is that of approaches to taxonomy development, i.e., focusing on the 'process' perspective of our work. In this perspective, we discuss how taxonomy development has been addressed so far in our domain.

2.1 Business Process Building Blocks

The concept of business process building blocks is a well-developed notion within the business process management domain, although the concept covers a range of definitions and a similar range of applications. In the context of our research, we focus on the identification of business process objects that capture the foundational properties of the process structure. Encapsulated as process elements [26], they enable the development of a component-based business process engineering technology. The approach aims to decomplex business processes into foundational process components and relatively simple dependency relationships [7].

Developing descriptions of characteristics of process components is required to identify the properties of these components with the purpose of advancing the use of these process 'parts', both intra-organizationally and inter-organizationally. General access to the process components is typically facilitated by the use of catalogs [29] or repositories [17]. In doing so, it is important to understand that describing these components can be performed with different points of view, or lenses, that heavily influence the choice of characteristics of the components. We observe three main lenses in research: the business operations lens, the information systems lens, and the information technology lens.

Today, a tight integration is required between the three lenses when building operational business systems. Non-alignment of the lenses is a major factor in the creation of complexity that hampers the capacity of organizations to deal with change. Adding to this complexity is the individual 'signature' in data management constructs that IT professionals unintentionally but inevitably impress on information systems solutions [11]. Too often, 'digital concrete' is the result that creates legacy structures that are heavily in the way of agility. Therefore, the nature of process building in terms of the used lens needs to be understood clearly. Next to this, the level of granularity of the process elements under consideration is an important issue [4].

Business process modelers are known to have a high degree of freedom in the decisions they have to make, and also when describing the business processes. This often leads to a lack of objectivity in business process models and descriptions: "model creation is more art than science and the resulting freedom can exacerbate the effective utilization of models. Process models are concise, selective, and arguably subjective representations because there is a lack of objectivity regarding terminology, perspectives, and granularity" [20]. This statement implies that existing approaches have their limitations in improving operational flexibility through the use of information technology. This is the case because they are either vendor-specific, with the lurking danger of locked-ins, are too abstract, demonstrate a partial solution, or do not support the transfer of operations over a variety of technology platforms. Many of the vendors position themselves more as productivity tools for extensions of Lean, TQM, etc. than as independent information architecture platforms [2, 5].

Given the impact of actual technology offerings on virtually all aspects of organizational functionality we infer that an integrated approach is required that covers the three discussed lenses, resulting in the meta-concept of integral business process engineering (BPE). At first glance, this may seem to be another endeavor to advocate the use of process building blocks as they are available in the marketplace at different levels of granularity, e.g., in the form of process patterns [1]. Our research direction differs at a fundamental level from these market offerings but also from theory directions [10] as our method focuses on the semantics of business process structures rather than on the syntax of these. We do not look to shape patterns of process elements but prefer to shape the content of operational activities rather than the form. We do this by identifying and analyzing the properties of process elements, using taxonomies to classify these to structure catalogs that enable the management of these information assets. In doing so, the semantics of business process activities are our starting point for analysis, comparison, and construction, rather than the structures of business processes, as advocated for example in work on process repositories [28].

2.2 Taxonomies

March and Smith [21] present four kinds of research contributions (artifacts) – constructs, models, methods, and instantiations – and two processes (research activities) – artifact building and artifact evaluation – that characterize design science research in IS. In this paper, we present our approach that is intended to support design researchers during their activities in developing a taxonomy for a specific domain. This method is an artifact that serves as a basis for future design science research, the purpose of which is to develop new taxonomies.

Taxonomy is described as "the scientific process of classifying things" (Oxford University Press, 2023). Mapping the properties of a collection of elements is done by using taxonomies as taxonomies offer an adequate framework for the organization of knowledge. Taxonomies "provide a structure and an organization to the knowledge of a field" [14]. They allow to postulate on the relationship between concepts [22] and constitute a fundamental mechanism for organizing knowledge [25].

The development and application of taxonomies have historically been implemented foremost in biology. However, taxonomies have been introduced to other fields of science more recently, such as manufacturing strategies [23] and information systems (IS). In BPM as a subdomain of IS, the ability to study relationships of and between the business process elements actors, activities, and objects is expected to be of value, which is why the development of taxonomies for these process elements is pursued in the scope of this research.

The method of developing taxonomies that are leading in our work is the method for taxonomy development presented by Nickerson et al. [19] and developed further by Kundish et al. [16]. The method for taxonomy development proposed by Nickerson et al. [19] allows the use of both empirical-to-conceptual and conceptual-to-empirical steps in the process of taxonomy construction: "The choice of which approach to use depends on the availability of data about objects under study and the knowledge of the researcher about the domain of interest" [19]. Important in the development of taxonomies is the choice of the domain of concern (the purpose of the analysis), its meta-characteristics, the dimensions of the taxonomy, and the determination of ending conditions. Typically, the development of taxonomies is done by iterations. If the ending conditions are met at the end of a development iteration, the taxonomy development comes to an end. On the other hand, if the determined ending conditions are not met at the end of a development iteration, a new iteration is initiated to advance the taxonomy. Kundish et al. [16] summarize the comments made on the contribution of Nickerson et al. [19] and stress the importance of a more pronounced way of expressing the purpose of a taxonomy as a basis for an improved evaluation mechanism. Their contribution results in a more articulated framework for taxonomy development as an expression of their 18 taxonomy development recommendations (TDRs).

3 Methodology

The method that we use for our taxonomy development is based on the work of Nickerson et al. [19] and Kundish et al. [16]. As discussed in the previous section, the latter is intended to be an extension of the former, so we take the work of Kundish et al. as the basis for our method. Kundish et al. describe a taxonomy development process consisting of 18 process steps organized in 6 phases: (1) identify the problem and motivate this; (2) define the objectives of a solution (the taxonomy); (3) design and develop the taxonomy; (4) demonstrate the taxonomy; (5) evaluate the taxonomy; (6) communicate the developed taxonomy. We basically use the same set of 18 steps as Kundish et al. but rearrange them in three ways.

Firstly, our taxonomy development takes place in collaboration with the organization that owns the business process that we use as empirical input for our development process. Despite the interest of this organization in our work, they cannot be involved in a highly iterative development process (they are not a research organization). For this reason, we rearrange the steps over the phases such, that there is a checking phase that can be performed within the research team on a frequent basis, and an evaluation phase that requires the involvement of the user organization, but on a less frequent basis. For this reason, we move Steps 13 and 14 from Phase V to Phase IV (and consequently relabel Phase IV).

Secondly, we feel that the process flow diagram in the work of Kundish et al. [16] is ambiguous in practice, as there is a single feedback loop with multiple 'entry' and 'exit' points. We remove this ambiguity by redefining the process flow as follows.

- We explicitly distinguish between an 'inner' feedback loop between Phases IV and III (corresponding with the frequent checking phase mentioned above) and an 'outer' feedback loop between Phase V and Phases I–III (corresponding to the infrequent evaluation phase mentioned above).
- We explicitly distinguish between the reasons to iterate from Phase V to one of the Phases I–III, depending on the observations in Step 17: problems with the users or purposes of the taxonomy, problems with the meta-characteristic or goals of the taxonomy, problems with the ending conditions of the development process, or problems with the elaboration of the dimensions or values of the taxonomy (i.e., the 'contents' of the taxonomy).

Thirdly, working with a large organization as our 'empirical source', we find that we have to pay explicit attention to not only defining the type of the objects that we classify in taxonomy development but also the scope of objects. In our case, we work with an organization that owns hundreds of complex business processes, which cannot all be an empirical basis for our work. Therefore, we have to explicitly scope our work within this organization to arrive at a feasible empirical basis. We reflect this in our method by splitting up the first step of the process of Kundish et al. into a type definition step and a scope definition step. To stay with step numbering of Kundish et al., we label our new steps as Step 1a and Step 1b.

These three modifications result in the process flow of our taxonomy development method as shown in Fig. 1. We consider this a 'practical' variation of the method process flow described by Kundish et al. [16]. For reasons of space limitations, we discuss the details of each of the steps in the application of the method in Sects. 4 and 5 of this paper.

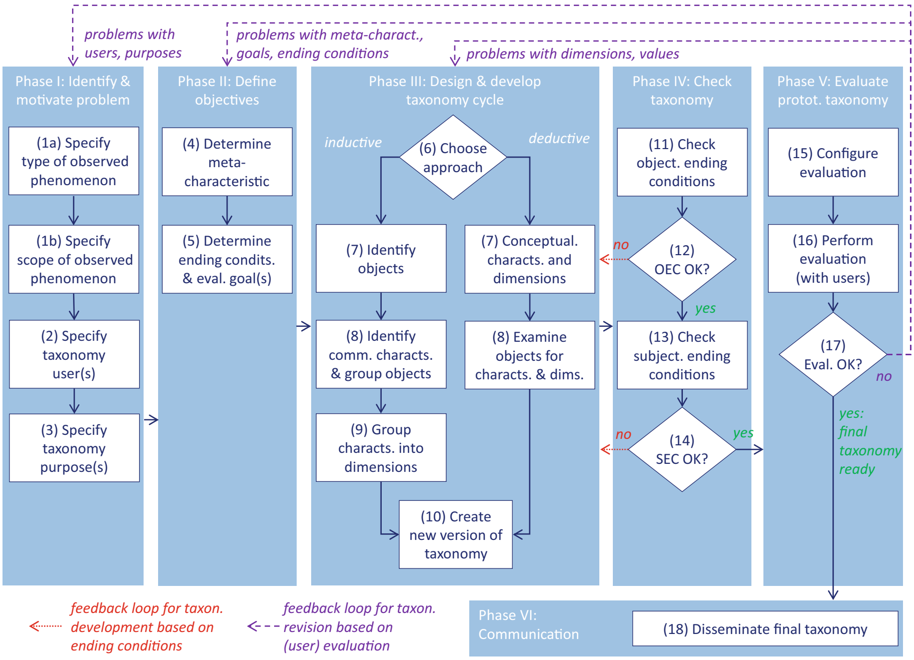


Fig. 1. The taxonomy development process (adapted from [19] and [16])

4 The Overall Taxonomy Development Process

In this section, we describe the overall taxonomy development process. This process is an instantiation of the process shown in Fig. 1, applied to the domain of activities for business process engineering. Below, we describe the process in pairs of phases that represent the setup of the development (Phases I and II), the iterative execution of the development (Phases III and IV), and the evaluation and dissemination of the development (Phases V and VI).

4.1 Phases I, II: Identify and Motivate Problem, Define Objectives

Following the (modified) guidelines of Nickerson et al. [19] and Kundish et al. [16], we perform Steps 1–5 in Phases I and II of Fig. 1 as follows per step:

1a. As the observed phenomenon, we specify business process activities (BPAs) as executed in business practice. We interpret BPAs through an information systems lens (as opposed to a business operations lens or an information technology lens).
1b. For the purpose of the work in this paper, we scope the observed set of BPAs to those executed by RWS in their LBS process (with the intention to broaden this scope later). The LBS process is a set of 8 sub-processes that is used to manage the Dutch national standard for building and using bridges in the road infrastructure. The LBS process contains 108 basic business process activities. As these contain many similar activities, we have abstracted them into 35 unique activities, or 'primitive activities' [26]. These 35 activities form the empirical set for the work covered in this paper.
2. As the taxonomy users, we take business process designers, i.e., designers of business processes from the information system perspective who work in industrial practice.
3. The purpose of the taxonomy is to provide a structure to define a set of generally reusable core BPA types from an activity content point of view. In the context of this paper, this set of reusable BPA types is scoped to RWS, but the intention is to in further work broaden the scope stepwise by analyzing more sets of BPAs.
4. The meta-characteristic of the objects under analysis is the nature of the core functionality of a BPA, isolated from the specifics of actors that execute a BPA and objects manipulated by a BPA. The overall approach (beyond this paper) is to develop taxonomies for these as well and then combine the three taxonomies.
5. We specialize the generalized ending conditions [19] into specific ending conditions as shown in Table 1.

Table 1. Overview of ending conditions (specialized from [19])

Objective ending conditions		Subjective ending conditions	
OC1	All BPAs in the set have been examined	SC1	The taxonomy is concise: it is parsimonious enough for practice
OC2	No BPAs in the set have been merged or split	SC2	The taxonomy is robust: it allows to differentiate between BPAs
OC3	All dimensions and values in the taxonomy are unique	SC3	The taxonomy is comprehensive: all BPAs can be classified
OC4	No new dimensions have been added to the taxonomy	SC4	The taxonomy is extendable: new dimensions or values can be added
OC5	No dimensions or values in the taxonomy have been merged or split	SC5	Taxonomy explanatory: the nature of BPAs is well explained

4.2 Phases III, IV: Design, Develop and Check Taxonomy

Following the guidelines of Nickerson et al. [19] and Kundish et al. [16], we perform Steps 6–14 in Phases III and IV of Fig. 1. As shown in the figure, the execution of these steps is performed in an iterative fashion: if either one of the objective ending conditions fails (Steps 11 and 12) or one of the subjective ending conditions fails (Steps 13 and 14) in Phase IV, Phase III is executed once more. This process repeats until all 10 ending conditions of Table 1 are satisfied.

In our taxonomy development process, we have used 7 iterations to develop a BPA taxonomy that satisfies all ending conditions. We discuss each of these iterations in detail in Sect. 5 of this paper. Each iteration has been performed with the two following additional considerations. Firstly, even though the development process flow (as in Fig. 1) states that not fulfilling a single objective ending condition implies executing an additional iteration (and hence evaluating the other ending criteria are irrelevant), we have evaluated all ending criteria in each iteration. As building a BPA taxonomy is new, we feel that this more holistic view on the process results in more learning about the structure of the domain of concern and hence a better result. Secondly, in our taxonomy development process, we classify BPAs from the activity perspective only, i.e., we do not take attributes in consideration of actors that execute a BPA or objects that are manipulated by a BPA. This brings a considerable level of abstraction to the classification task. Our aim is to execute similar taxonomy development processes for actors and objects in business processes, such that we can combine the three taxonomies into a three-dimensional classification space of what we call primitives, i.e., activity-actor-object triplets that can be used as standard building blocks in business process engineering [26].

4.3 Phases V, VI: Evaluate Prototype Taxonomy, Disseminate Final Taxonomy

Following the guidelines of Kundish et al. [16], we perform Steps 15–18 of Phases V and VI of Fig. 1. The main effort so far has been devoted to setting up (Step 15) and executing (Steps 16 and 17) the evaluation of the developed taxonomy, as explicitly stipulated by Kundish et al. [16], as detailed below. We are currently in the process of organizing the dissemination of the developed taxonomy (Step 18).

As working with taxonomies is completely new for the intended taxonomy users at RWS, we decided to keep the first round of evaluation fairly informal. Using an approach like TAM with novices is considered to be an overkill for the current phase of the project. We focused the evaluation (more or less in ‘tuned down’ TAM [9] style) on the following quality aspects of the developed taxonomy and related questions:

- Understandability: do the professionals at RWS understand the concepts, dimensions and values of the developed taxonomy?
- Completeness: are all essential elements of the RWS LBS process covered by the developed taxonomy?
- Usability: can the taxonomy be used in RWS practice, does it have a digestible complexity?
- Intention to use: if the taxonomy is further developed, would RWS use the taxonomy-based approach to streamline their processes?

To evaluate the taxonomy along these lines, two informal evaluation sessions have taken place: one session with all stakeholders involved in the process to obtain an overall impression of the opinion about our work, and one in-depth session with a business process engineer. In the first session, we presented the taxonomy-building approach and an overview of the taxonomy as a result of this. The overall opinion towards our approach and the developed taxonomy was positive, even though this is a completely new take on business process management at RWS. In the in-depth session, the developed taxonomy was presented in detail. The RWS process engineer did not have any corrections or additions to the taxonomy. As this is a new approach towards process analysis and construction at RWS, this was not very surprising to us: the amount of structure and detail may still be overwhelming. Our approach is seen as promising for managing their business processes, specifically from the aspect of aligning the definitions or related processes.

Evaluating the quality aspects raised above, we come for now to the following conclusions. For understandability, we have reached a basis, but work has to be done yet. This is to a large part attributable to the novelty of the approach to RWS. For completeness, according to our evaluation, our taxonomy is perceived to be complete. Taking the previous point into consideration, however, we will need to reiterate completeness at a later point in our project. To increase usability, the taxonomy as developed will need additional explanatory elements and training of users to become usable in practice without our intervention. We plan to address this in a later stage of our work. With respect to intention to use, RWS shows an intention to use our approach (given that the above points receive proper attention), as the issues addressed by the approach are of great importance to the organization (as briefly explained in the introduction of this paper).

Given these preliminary evaluation results, we do not have a basis yet to iterate over Phases I-IV at RWS, following the recommendations of Kundish et al. [16]. As we heavily value the relevance of our work, we rather do not perform a 'pro forma' iteration to 'appear more complete'. As we discuss in the concluding section of this paper, we will continue our current work with a next large case study, which will provide the basis for further development and a more in-depth evaluation of the developed taxonomy. We expect that this will provide a more complete basis to perform one or more iterations over the development process, thereby completing the entire development process.

5 The Taxonomy Building Iterations

In this section, we describe the iterative process of taxonomy building, detailing the execution of Phases III and IV of the overall approach (see Fig. 1), of which the outline has been discussed in Sect. 4.2.

5.1 Iteration 1

In the first iteration of the process of taxonomy building, we choose a deductive approach to create a basis. As we work in the information processing domain, we construct two dimensions that together form an extension of the well-known CRUD typology of database manipulations. The *data creation* dimension has two values: *data acquisition* and *data generation*. This dimension allows distinguishing between two sources of data for activities. The *data use* dimension has four values: *process*, *read*, *update*, and *delete*. We add the *process* value to classify activities that have a complex information processing character (such as revising a document), whereas *update* refers to simple information manipulation (such as entering data in a form). Note that we use two dimensions as a business process activity may both create data and use data of a different type, so needs to be classified in both dimensions.

Table 2 shows a fragment of the application of the two dimensions on our empirical data set from the RWS case. Note again that the entire set consists of 35 BPAs. We show a fragment in each iteration because of space limitations – the complete data sets are available in an online appendix at: https://drive.google.com/file/d/1PNbmoXmMmXoELesSueURRNJxfHU3HR2K/view?usp=sharing.

Table 2. Data sample from Iteration 1

dimension	data creation		data use			
value	data acquisition	data generation	process	read	update	delete
send	X			X		
submit	X				X	
publish		X	X			
inform		X		X		
file		X			X	
receive	X			X		
update	X				X	
register		X			X	
forward		X	X			
enter	X		X			

As shown in the table, the *data creation* and *data use* dimensions show a reasonable spread of BPAs over the values of the dimensions. The exception is the value *delete*, which is not used (also in the entire data set). As we expect this value to be required at a later stage or in a broader taxonomy context, however, we have decided to keep this value for the time being.

We have added new dimensions in this iteration, so objective ending condition OC1 has not been met. The taxonomy is not considered sufficiently concise, robust and explanatory, so subjective ending conditions SC1, SC2, and SC5 have not been met either (note that these are indeed subjective assessments, as explicitly explained in [19]). Hence, we have to perform a next iteration, because of both objective and subjective ending conditions.

5.2 Iterations 2 and 3

In Iteration 2 of our development process, we first have deductively chosen the dimension *data storage*, with values *record*, *file*, and *database*. The intention of this dimension is to classify activities with respect to the way they store data. As we classify *activities* only (i.e., without explicitly considering the objects that they may store), there appeared to be too little context to make a proper classification in this dimension, and hence this dimension does not add towards ending conditions SC2 and SC5. We therefore have decided to reject this dimension.

In Iteration 3, we have replaced dimension *data storage* by dimension *data move*, with values *none*, *inter-company*, and *intra-company*. This deductively chosen dimension enables classifying activities with respect to the scope in which they move (send) data to other activities. Table 3 shows a fragment of the application of the new dimension to our empirical data set. This dimension appears to be useful in classifying data. The table illustrates that even in this fragment of the data set, all values of the dimension occur.

Table 3. Data sample from Iteration 3 (Dimension 1 not shown)

dimension	data use				data move		
value	process	read	update	delete	none	inter cy	intra cy
send		X					X
submit			X				X
publish	X					X	
inform		X					X
file			X				X
receive		X				X	
update			X		X		
register			X		X		
forward	X						X
enter	X				X		

We have added a new dimension to the taxonomy, so OC4 is not met. Also, SC1, SC2 and S5 are not met to our satisfaction (which is, as prescribed by the method, again a subjective evaluation). Hence, we have decided to execute a next iteration.

5.3 Iteration 4

In Iteration 4, we have analyzed deductively the way an activity (or the actor executing an activity) performs its messaging to other activities. This has resulted in dimension *messaging mode* with values *physical* and *automated*. Table 4 shows a fragment of the application of the new dimension to the empirical data set. The dimension shows to be useful, so we have accepted it as part of the taxonomy under construction.

In this iteration, we have added a new dimension to the taxonomy, so OC4 is not met. Also, SC1, SC2 and S5 are still not met. Hence, we have decided to execute a next iteration.

5.4 Iteration 5

In formal information processing, data validation is of great importance. This is certainly the case in the RWS processes that we analyze, as they are related to standardization. Therefore, taking an inductive approach, we have chosen *data validation* as the next dimension to add to our taxonomy. In this dimension we classify activities with respect to the goals of data validation: *consistency* with other data, *conformance* with respect to internal constraints, or *compliance* with externally imposed regulations. Table 5 shows a

Table 4. Data sample from Iteration 4 (Dimensions 1–2 not shown)

dimension	data move			messaging mode	
value	none	inter cy	intra cy	physical	automated
send			X	X	
submit			X		
publish		X			X
inform			X	X	
file			X		X
receive		X		X	
update	X				X
register	X			X	
forward			X		X
enter	X				X

fragment of the application of the new dimension to our empirical data set. To illustrate how all values are used, we show different rows of our data set than in Table 4.

Table 5. Data sample from Iteration 5 (Dimensions 1–3 not shown, different rows shown)

dimension	messaging mode		data validation		
value	physical	automated	consistency	conformance	compliance
update		X		X	
register	X		X		
forward		X		X	
enter		X		X	
start		X		X	
prepare	X		X		
apply		X			X
analyze	X		X		
decide	X		X		
complete		X		X	

In this iteration, we have added another new dimension to the taxonomy, so OC4 is again not met. We have felt that the current taxonomy, with its five dimensions, is concise, so meets subjective ending condition SC1. However, SC2 and S5 are still not met. Hence, we have decided to execute yet another iteration.

5.5 Iterations 6 and 7

In this sixth iteration, we have chosen to take an inductive approach: when looking at activities, we see that some are meant to change the form (format) of data, some are not. Some change the content of data (e.g., edit a document), some don't. Some are meant to reduce the size of data (e.g., produce a summarization of a data set or document), some keep the size intact. This has led to three new dimensions: *form preservation*, *content preservation*, and *size preservation*. They all have the values *yes* and *no*, indicating whether they preserve the characteristic of the data addressed by the respective dimension.

Table 6. Data sample from Iteration 6 (Dims. 1–4 not shown)

dimension	data validation			form preservation		content preservat.		size preservation	
value	consist.	conform.	compl.	yes	no	yes	no	yes	no
update		X		X		X		X	
register	X			X		X		x	
forward		X		X		X		X	
enter		X		X		X		X	
start		X		X			X		X
prepare	X				X		X		X
apply			X		X		X		X
analyze	X			X		X		X	
decide	X			X		X		X	
complete		X			X		X		X

Table 6 shows a fragment of the application of the three new dimensions to our empirical data set. We use the same subset of data as in Table 5 to illustrate the occurrence of all values.

In this iteration, we have added three new dimensions, so OC4 is not met for sure. Given the protocol of Fig. 1, we had to perform a next iteration.

In Iteration 7, we have concluded that no structural changes are necessary to the taxonomy at this point, so the taxonomy meets all objective ending conditions. We also have felt that all subjective ending conditions were met. With the three new dimensions of the previous iteration, the taxonomy has become adequately robust and explanatory for its intended use. Hence, we have ended the iteration cycle with a (for the scope of this paper) complete taxonomy comprising of eight dimensions with each between two and four values, as shown in Table 7.

Table 7. Overview of developed taxonomy structure

Dimensions	Values			
Data creation	Data acquisition	Data generation		
Data use	Process	Read	Update	Delete
Data move	None	Inter-company	Intra-company	
Messaging mode	Physical	Automated		
Data validation	Consistency	Conformance	Compliance	
Form Preservation	Yes	No		
Content Preservation	Yes	No		
Size preservation	Yes	No		

6 Reflection

Our presented work on taxonomies is based on a rich real-world environment (RWS) and thus offers a strong empirical basis for contributions to the use of taxonomies as instruments to structure knowledge. In executing our work, we have made several interesting observations in this context, related to the domain of concern for which we develop our taxonomies: business process engineering.

Firstly, using the Nickerson framework [19], we found that our results benefit from an explicit definition of the 'domain of concern' as an important element of the structure. The identification of this domain of concern as a first step in developing taxonomies provides context and meaning to the definition of the meta-characteristics and subsequently to the specification of taxonomy dimensions, their values and ending conditions for the development process. We observed that developing a taxonomy of process activities for the domain of information systems results in different meta-characteristics, compared to the domain of business operations or information technologies. As noted before, we refer to this definition of the domain of concern as the 'lens' used in analyzing a collection of objects.

Secondly, the characteristics of the collection of objects under consideration are of importance as they influence the applicability and contribution of a taxonomy. This point has been raised previously [18]. In our pilot cases, we found that differences in the character (e.g., formalized vs. non-formalized) and the granularity of the business process objects under observation [4] had a profound impact on the resulting taxonomy and its match with its purpose. Even though we think that we have made adequate progress here, this issue requires further attention in our work.

Thirdly, the BPE domain and the IS lens are rather abstract in nature – compared to other domains where taxonomies are popular, such as biology, where objects under analysis are more 'physical'. We found that this appears to lead to a bias towards deductive iterations as a basis in the taxonomy development process (our first four iterations are fully deductive). Inductive iterations are harder – perhaps because abstract objects are harder to observe than physical ones – but are required to achieve a good 'fit' of a taxonomy with the domain of concern it is intended for.

Finally, taxonomies are the result of a series of analysis iterations and iterations are evaluated on an individual basis either to be accepted or rejected. Next to this iterative evaluation process during construction cycles, the resulting taxonomy as a whole needs to be evaluated (in Step 16 of Fig. 1). Here we agree with the observations of Kundish et al. [16], but further operationalize the recommendations made in this work. In the case study described in this paper, this evaluation has been executed in a rather informal way. In the next case study of the research project, we plan to address this more formally, applying structures in methods like TAM [9] much more rigorously.

7 Conclusions and Further Work

With the work described in this paper, we have continued our design science research effort to develop a method that enables a mass customization approach to business process construction. After our initial quest into the properties of atomic building blocks, the levels of syntactic abstraction involved and the redundancies demonstrated [26], this paper focuses on the semantic expression of one of the three individual process factors (activities, actors and objects) that constitute the foundational components of business process elements. It is important to observe that the developed taxonomy approach is not the end goal of our research endeavor but is instrumental to our quest towards what we call *business process DNA*, as mentioned in the introduction to this paper, and as a fundamental element in business process engineering.

We see our contributions to the BPM research domain as twofold. Firstly, our work is among the first to use the promising research direction of taxonomy development for business process analysis and design in the context of complex, real-world business process management. To guarantee the relevance of our work in terms of design science research [15, 25], we explicitly choose this practical anchoring with a strong empirical basis. We guarantee rigor [15, 25] in our work by embedding our work in relevant approaches and using strict method execution. Secondly, our work provides a strong basis for a process activity semantics structure. This complements the existing work on process syntax structure, e.g., on process patterns.

In our next business case (which is on the way at the time of finishing this paper), we will broaden our taxonomy scope by developing taxonomies for the other two process factors (actors and objects) and broaden our view to include a business operations perspective next to an information systems perspective. We will perform this work in the industrial environment of a large construction company aiming at assisting them in aligning their business processes for physical asset management and deployment with their digital asset management processes.

Acknowledgments. The colleagues at Rijkswaterstaat (RWS) are thanked for their positive cooperation in the execution of the research reported in this paper. Jan Veraart of TUE is thanked for his assistance in the analysis of the RWS case.

References

1. Van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distrib. Parallel Databases **14**, 5–51 (2003). https://doi.org/10.1023/A:1022883727209
2. Van der Aalst, W.: Process Mining, Data Science in Action. Springer, Berlin (2016). https://doi.org/10.1007/978-3-662-49851-4
3. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. Oxford University Press, Oxford (1977)
4. Beerepoot, I., et al.: The biggest business process management problems to solve before we die. Comput. Ind. **146**, 103837 (2023). https://doi.org/10.1016/j.compind.2022.103837
5. vom Brocke, J., Rosemann, M. (eds.): Handbook on Business Process Management 2. IHIS, Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45103-4
6. Vom Brocke J.: towards process science: embracing interdisciplinary research opportunities in the digital age. In: Keynote Speech BPM Conference, Seville, Spain (2020)
7. Chang, J.F.: Business Process Management Systems. Auerbach Publications, Boca Raton (2016)
8. Ching, F.: Architecture, Form, Space & Order. John Wiley, New York (1996)

9. Davis, F.: A technology acceptance model for empirically testing new end-user information systems: theory and results. Sloan School of Management, Massachusetts Institute of Technology (1986)
10. Dijkman, R., Vanderfeesten, I., Reijers, H.: Business process architectures: overview, comparison and framework. Enterpr. Inf. Syst. **10**(2), 129–158 (2016). https://doi.org/10.1080/17517575.2014.928951
11. Dumas, M., LaRosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Berlin (2017). https://doi.org/10.1007/978-3-642-33143-5
12. Erens, F.: The Synthesis of Variety: Structuring Product Families. Eindhoven University of Technology, Eindhoven (1996)
13. Fellmann, M., et al.: A taxonomy and catalog of business process model patterns. In: Proceedings 22nd European Conference on Pattern Languages of Programs, Article No. 19, pp. 1–10. ACM (2017). https://doi.org/10.1145/3147704.3147725
14. Glass, R., Vessey, I.: Contemporary application-domain taxonomies. IEEE Softw. **12**(4), 63–76 (1995). https://doi.org/10.1109/52.391837
15. Hevner, A., Chatterjee, S.: Design Research in Information Systems. Springer, New York (2010)
16. Kundish, D., et al.: An update for taxonomy designers: methodological guidance from information systems research. Bus. Inf. Syst. Eng. **64**(4), 421–439 (2022). https://doi.org/10.1007/s12599-021-00723-x
17. Lincoln, M., Gal, A.: Searching business process repositories using operational similarity. In: Proceedings on the Move to Meaningful Internet Systems 2011, pp. 2–19 (2011). https://doi.org/10.1007/978-3-642-25109-2_2
18. Mwilu, O., Prat, N., Comyn-Wattiau, I.: Taxonomy development for complex emerging technologies. In: Proceedings 19th Asia-Pacific Conference on Information System (2015)
19. Nickerson, R., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. Eur. J. of Inf. Syst. **22**, 336–359 (2013). https://doi.org/10.1057/ejis.2012.26
20. Mannaert, H.: Normalized systems: towards designing evolvable modular structures. In: Proceedings 7th International Conference on Software Engineering Advance (2012)
21. March, S., Smith, G.: Design and natural science research on Information Technology. Decis. Supp. Syst. **15**(4), 251–266 (1995). https://doi.org/10.1016/0167-9236(94)00041-2
22. McKnight, D., Chervany, N.: What trust means in e-commerce customer relationships. Int. J. Electron. Commer. **6**(2), 35–59 (2001). https://doi.org/10.1080/10864415.2001.11044235
23. Miller, J., Rohm, A.: A taxonomy of manufacturing strategies. Manag. Sci. **40**(3), 285–304 (1994)
24. Oosterhout, M. van: Business Agility and Information Technology in Service. ERIM PhD Series in Research in Management 198. Erasmus University, Rotterdam (2010)
25. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. Man. Inf. Syst. **24**(3), 45–77 (2007). https://doi.org/10.2753/MIS0742-1222240302
26. Soetekouw, A., Grefen, P., Turetken, O., Vanderfeesten, I.: A mass customization approach to business process modularization. In: Proceedings IEEE 23rd Conference on Business Informatics, pp. 29–38. Springer, Heidelberg (2021). https://doi.org/10.1109/CBI52690.2021.10052
27. Steen, M., Akehurst, D., ter Doest, H., Lankhorst, M.: Supporting viewpoint-oriented enterprise architecture. In: Proceedings 8th IEEE International Enterprise Distributed Object Computing Conference, pp. 201–211. IEEE, Piscataway (2004)

28. Weber, B., Reichert, M., Mendling, J., Reijers, H.: Refactoring large process models. Comput. Ind. **62**, 5 (2011). https://doi.org/10.1016/j.compind.2010.12.012
29. Zhang, Z., et al.: A novel complex manufacturing business process decomposition approach in cloud manufacturing. Comput Ind. Eng. **144**, 106442 (2020)

Modeling in an Enterprise Context

A Taxonomy for Platform Revenue Models: An Empirical-to-Conceptual Development Approach

Nedo Bartels[1(✉)], Matthias Koch[1], Anna Schmitt[1], and Jaap Gordijn[2,3]

[1] Fraunhofer IESE, Kaiserslautern, Germany
{nedo.bartels,matthias.koch,anna.schmitt}@iese.fraunhofer.de
[2] The Value Engineers, Soest, The Netherlands
jaap@thevalueengineers.nl
[3] Vrije Universiteit, Amsterdam, The Netherlands
j.gordijn@vu.nl

Abstract. In the field of Information Systems and Software Engineering, taxonomies are widely employed to organize and present well-designed knowledge. They play a crucial role in identifying relevant dimensions and characteristics associated with the objects under study. This paper focuses specifically on revenue models for platform business models, which facilitate the connection between providers and consumers in two-sided markets. For example, the Vinted Marketplace charges a transaction-based fee of 5% for each item sold, while nebenan.de offers platform access for a monthly subscription fee. Although these revenue model types differ, they both lead to distinctive and successful revenue models. Understanding and formalizing these revenue mechanisms is fundamental for the systematic design of revenue models for platform business models. This paper follows a proven taxonomy development method with two empirical-to-conceptual iteration cycles involving seven use cases. It introduces a comprehensive taxonomy comprising 15 dimensions and 79 characteristics. The proposed taxonomy contributes to the formalization of revenue models for platform business models and enhances the current understanding of the monetization strategies used by digital platforms to generate revenues. This paper supports researchers and practitioners involved in the design process of platform business models.

Keywords: Platform Business Model · Revenue Model · Taxonomy · Digital Platform

1 Introduction

The relevance of digital platforms is increasing continuously, and many companies have created new business models based on platforms, e.g., Uber Ride, Airbnb Lodging, Spotify Music, and eBay Marketplace. Companies are affected

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 189–205, 2024.
https://doi.org/10.1007/978-3-031-46587-1_11

by this platform trend and need to make strategic decisions about how to position their business model to gain competitive advantages [19]. For example, Salesforce created AppExchange, a marketplace for B2B applications, as part of the redesign and expansion of their existing services. Defining a company's business model is one way to describe the underlying logic of a business, and can be described based on three dimensions [23]: (1) What values, products, or services does a company offer for its customers (value delivery)? (2) How does a company create its values, products, or services (value creation)? (3) How does a company generates its revenue (value capture)? It is essential for a business to clarify these questions in order to define the company's strategy, how it aims to create value, and how it can capture value. Digital platforms are constantly emerging, and it is crucial for companies to pay attention to them in order to understand how platform business models can create and capture value. We argue that there is a lack of guidance for identifying suitable platform business models and designing effective revenue models to monetize digital platforms. To achieve this, we propose a taxonomy in this paper, which classifies dimensions and characteristics of revenue models specific to these platform businesses. Based on this, we derive the following research question: *What dimensions and characteristics can be used to describe revenue models of platform business models?*

To answer this research question, this paper follows a taxonomy development process using an empirical-to-conceptual approach according to Nickerson et al. [18]. Based on our previous taxonomy [7], we conducted two iteration cycles, analyzed seven platform business models, and extracted 26 revenue model types. The result of this study is a revised taxonomy comprising 15 dimensions and 79 characteristics for revenue models of platform business models and a classification of seven use cases. This paper is structured as follows: In Sect. 2, we introduce key terms and relevant related work. Sect. 3 presents the research design of the taxonomy development process, and Sect. 4 presents the findings, including the taxonomy we created in Sect. 4.1, the documented changes from both empirical-to-conceptual iteration cycles in Sect. 4.2, and the presentation of the analyzed platform revenue models in Sect. 4.3. Finally, Sect. 5 comprises our discussion, limitations, and future work.

2 Theoretical Background

A business model represents the underlying logic of a business, with a focus on how economic value is created, distributed, and consumed in a network of actors that are organizations [13]. In our case, we are looking at digital platforms and their business models, as they bundle several actors via a digital platform, and call this construct a platform business model. We advocate the logic that, e.g., the Uber Ride platform operator (asset broker) brokers rides (assets) provided by drivers (asset providers) for passengers (asset consumers) on its digital platform, as a software system that serves as the technical foundation, and is defined as a digital ecosystem according to [14]. In our understanding, a *platform business model* can be described with the following characteristics adapted from the definitions of [13,14,17,22]: (1) A platform business model describes the concept of how economic value is created, distributed, and consumed in a *network of parties*, called a digital ecosystem. (2) It creates value through *a digital platform*, operated by a platform operator (i.e., asset broker), which connects at least *two market sides* - asset providers and asset consumers. (3) It *brokers assets* such as products or services via its digital platform. (4) A digital platform serves as the *hub of a digital ecosystem* consisting of companies working collaboratively and competitively to meet customer needs. The *revenue model* is part of the value capture dimension of a business model and clarifies which monetization mechanisms are used to capture value from the platform's mediation activities between its two-sided markets [6]. As each platform business model creates value differently, different revenue model types are needed to capture value. A revenue model should define appropriate revenue sources and revenue streams to capture the value delivered [22].

In this paper, we develop a taxonomy to classify dimensions and characteristics of revenue models for platform business models. A *taxonomy* is a form of classifying and grouping concepts or objects, whether derived from empirical evidence or conceptual frameworks. It provides researchers and practitioners with a means to analyze, structure, and comprehend complex domains [18]. Various taxonomies have been proposed in the literature to conceptualize digital platforms and their business models holistically. Van de Ven et al. [24] developed a *taxonomy for business models of data marketplaces*, which includes the five dimensions, 'revenue model', 'pricing model', 'price discovery', 'smart contract', and 'payment currency'. Springer and Petrik [20] proposed a *taxonomy for platform pricing in the context of the Industrial Internet of Things (IIoT)*, identifying 'pricing model', 'subsidization', and 'pie-splitting' as relevant impact factors for a revenue model. Staub et al. [21] presented a *taxonomy for digital platforms*, focusing on 'key revenue stream', 'price discovery', and 'price discrimination' as relevant dimensions for a revenue model. Freichel et al. [12] introduced a *taxonomy for digital platforms* categorized under 'technological perspective', 'economic perspective', and 'socio-cultural perspective'. Here, 'pricing mechanism' and 'primary revenue source' are specified. Täuscher and Laudien [22] proposed a *taxonomy for marketplace business models*, highlighting four dimensions for value capture, i.e., 'key revenue stream', 'pricing mechanism', 'price discrimination', and 'revenue source'.

While these existing taxonomies provide a comprehensive understanding of digital platforms and their business models, they do not specifically focus on revenue models for platform business models. The literature lacks a universal understanding, as authors mention similar dimensions (e.g., 'key revenue stream' used by Staub et al. [21] and Täuscher and Laudien [22]), while others introduce additional ones (e.g., 'payment currency' by Van de Ven et al. [24]). The lack of a taxonomy that reflects common dimensions and characteristics highlights a gap in the literature on formalizing revenue models of platform business models. Our research aims to address this gap by exploring and categorizing revenue model types for platform business models, thereby contributing to a better understanding of how digital platforms generate revenue. This knowledge can be used in the future to provide tool support, assisting practitioners in designing their own monetization strategies for platform business models.

3 Research Design

In the development of our taxonomy, we followed the guidelines proposed by Nickerson et al. [18]. These guidelines are widely recognized in the fields of Information Systems and Software Engineering, having proven their effectiveness in structuring existing knowledge about digital platforms and business models (as demonstrated, among others, in the taxonomy development of Staub et al. [21], Van de Ven et al. [24], or Weking et al. [25]). Although Kundisch et al. [16] have extended the approach of Nickerson et al. with their work on taxonomy evaluation, in this paper, we employed the taxonomy building methodology of Nickerson et al. [18]. Nonetheless, there is potential to enrich this research design by integrating the taxonomy evaluation methodology proposed by Kundisch et al. [16] in the future. The guidelines of Nickerson et al. [18] provide two approaches for developing a taxonomy: empirical-to-conceptual and conceptual-to-empirical. Building upon the initial taxonomy by Bartels et al. [7], which employed a conceptual-to-empirical approach, we present in this paper a revised version of the initial taxonomy. The initial taxonomy is depicted in Fig. 1. In this paper, we followed an empirical-to-conceptual approach in two iterations. The taxonomy development process, shown in Fig. 2, consists of three iteration cycles. In the first step of the taxonomy development process, the object of the taxonomy and its ending conditions were defined. The initial taxonomy (Fig. 1) as a result of *iteration 1* serves as the starting point for iterations 2 and 3. Iteration 1 is mentioned in Fig. 2 to provide a comprehensive overview of the research design. However, it is not discussed in detail in this paper. For more information on iteration 1, see Bartels et al. [7]. In *iteration 2*, we validated the practical relevance of the taxonomy by applying it to 19 revenue model types from five existing platforms mainly operated in Germany. This extended our analysis as we mapped the revenue models onto the initial taxonomy and revised it in a second version. The data presented in this paper is fully documented and available in [8]. In *iteration 3*, we further refined the taxonomy by applying it to seven revenue model types from two research projects until all ending conditions were met. In total, we applied the finalized taxonomy to 26 revenue model types.

3.1 Determination of Meta-characteristics and Ending Conditions

We aim to create a taxonomy that includes the main dimensions and characteristics of revenue models of platform business model. For this, we defined our relevant revenue model configuration aspects as our *meta-characteristics*, like the revenue source and the revenue stream of a digital platform. To be accepted, the taxonomy must meet both objective and subjective *ending conditions* according to Nickerson et al. [18]: The taxonomy should (1) include the *main dimensions*

Revenue model			Revenue model characteristics of a platform business model							
Revenue model of the asset broker	DB1	Revenue model type of the asset broker	Access model	Commission model	Pay per use model	Sales model	Advertising model	Listing model	Donation model	Other
	DB2	Revenue stream of the asset broker	Access fees to platform	Commission fees on platform transactions	Commission fees on usage	Sales model of platform services	Advertising fees for space	Listing fees on platform	Donations	Other
	DB3	Revenue source of the asset broker	Asset consumers		Asset providers			Third party		Other
	DB4	Payment frequency of the platform price	One-time		Subscription-based frequency			Usage-based frequency		Other
	DB5	Price discovery of the platform price	Platform price set by asset providers		Platform price set by asset consumers		Platform price set by negotiation		Platform price set by asset broker	Other
	DB6	Price mechanism of the platform price	Fixed platform pricing				Variable platform pricing			Other
	DB7	Price discrimination of the platform price	Feature-based price discrimination		Quantity-based price discrimination			Location-based price discrimination		Other
Revenue model of the asset provider	DP1	Revenue model type of the asset provider	Sales model		Rental model			Pay per use model		Other
	DP2	Revenue stream of the asset provider	Sales of assets		Rental fees for assets			Usage fees for assets		Other
	DP3	Revenue source of the asset provider	Asset consumers		Asset broker			Third party		Other
	DP4	Payment frequency of the asset price	One-time		Subscription-based frequency			Usage-based frequency		Other
	DP5	Price discovery of the asset price	Asset price set by asset providers		Asset price set by asset consumers		Asset price set by negotiation		Asset price set by asset broker	Other
	DP6	Price mechanism of the asset price	Fixed asset pricing				Variable asset pricing			Other
	DP7	Price discrimination of the asset price	Feature-based price discrimination		Quantity-based price discrimination			Location-based price discrimination		Other

Fig. 1. Initial taxonomy of the first iteration [7]

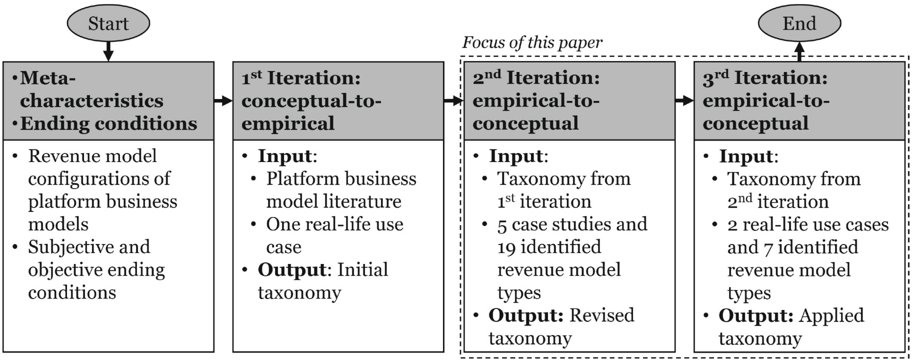


Fig. 2. Taxonomy development process adapted from Nickerson et al. [18]

and characteristics of revenue models for platform business models, and (2) not incorporate *new dimensions or characteristics* in the last iteration. Furthermore, the taxonomy must (3) strike a balance between being *meaningful and not being too complex or overwhelming*, and (4) also be *extensible* to accommodate new dimensions or characteristics. Lastly, (5) each dimension and characteristic should offer *explanatory* value about platform revenue models.

3.2 First Iteration: Conceptual Development

The detailed description of iteration 1 can be found in the research data [8], and is outside the scope of this paper. A literature review on revenue models of platform business models was conducted using several databases, resulting in 930 papers. Out of these, 34 papers were deemed relevant and used to develop the taxonomy. Exclusion criteria were applied to the remaining 896 papers. A full-text review of the 34 included papers led to the identification of 68 dimensions and 258 characteristics for revenue models of platform business models. We synthesized the data by creating a concept matrix that summarizes the classifications for revenue models of platform business models in eight dimensions. To further develop and refine the extracted taxonomy of revenue models of platform business models, a UML class model was created. The aim of this model is to express the relationships between the relevant dimensions and their respective characteristics, and to specify the taxonomy derived from the literature review. Based on this, and to ensure applicability, the taxonomy was applied to a real-life use case from a research project. The resulting taxonomy is shown in Fig. 1.

3.3 Second Iteration: Empirical Development

To address the empirical relevance of the taxonomy, we conducted a desk research between January and April 2023 to extract data. As a primary source of empirical cases on platform business models, we used the work of Koch et al. [15], which includes a list of 43 described platform business models. To select relevant platform business model cases, we assessed whether (1) two market sides, i.e., asset provider and asset consumer, could be identified; (2) sufficient information on the revenue model was available to understand the logic of value capture; and (3) the platform business model was not too complex and had no complex interdependencies with other related business models (see, for instance, Amazon Prime and its video, marketplace, and delivery connections). Through the analysis, we discovered that a complete revenue model for a platform business model cannot always be represented by a single revenue model type, but may involve combinations of several revenue model types. For example, the Vinted platform employs both a commission model for each transacted item and generates revenues through the sale of additional platform services, representing two distinct revenue model types. Consequently, we identified 19 revenue model types across five platforms: Tyre24, empto, MyHammer, Vinted, and nebenan.de, and applied the initial taxonomy to each one individually. Here, each revenue model type was mapped onto the taxonomy to assess whether it could be fully captured, and any missing dimensions or characteristics were documented in an Excel taxonomy grid. Each identified discrepancy or gap was then marked within the taxonomy and documented as a comment. Afterwards, all documented changes and comments were aggregated and, reviewed, and a revised version of the taxonomy was created. The entire process of the second iteration is documented and can be found in the research data [9].

3.4 Third Iteration: Application in Real-Life Use Cases

In the third iteration, two research projects on digital platforms were used as case studies, and their revenue model descriptions were extracted from internal project documents. The revised taxonomy created after iteration 2 was used to identify and classify seven different revenue model types. The taxonomy was initially applied to one project, and any gaps were documented and addressed before it was used it on the second project. In the second research project, no gaps or changes were identified. The taxonomy had fulfilled all the necessary ending conditions in the second research project, meaning the taxonomy was finalized and the development process was stopped.

3.5 Selected Use Cases

For iteration 2 of our taxonomy development process five different platform business models were examined. We, started with Tyre24 [4] and then analyzed those of empto [1], MyHammer [2], Vinted [5], and nebenan.de [3]. To validate the completeness and correctness of the developed taxonomy in iteration 3, two research projects dealing with digital platforms were used as real-life use cases.

1) *Tyre24* is a digital platform for car parts trading. The Tyre24 platform is operated by the Saitow company (asset broker), which brokers car parts, such as tires (assets) provided by suppliers and distributors of car parts (asset providers), to car repair shops (asset customers).
2) *empto* is a digital platform for companies to manage their waste. The empto platform is operated by the Zentek Services company (asset broker), which brokers waste disposal services, such as disposal of glass waste (assets) provided by professional waste disposers (asset providers) to waste producing companies (asset customers).
3) *MyHammer* is a platform for finding local skilled trade businesses. The MyHammer platform is operated by the MyHammer company (asset broker), which brokers craft services, such as home repair and renovation services (assets) provided by local skilled trade businesses (asset providers) to homeowners (asset customers).
4) *Vinted* is a platform for buying and selling second-hand clothing and accessories. The Vinted platform is operated by the Vinted company (asset broker), which brokers clothing items, such as t-shirts (assets) provided by individual sellers (asset providers), to buyers (asset customers).
5) *nebenan.de* is a social network platform for local communities to connect and exchange goods and information. The nebenan.de platform is operated by the Good Hood company (asset broker), which brokers neighborhood-related information (assets) provided by local individuals, businesses, and organizations (asset providers) to neighbors (asset customers).

6) The first research project, called *Smarte.Land.Regionen* (SLR), was previously explored by Bartels and Schmitt [10] and aims to enhance public services in rural areas through digital solutions. In this paper, the developed taxonomy and the insights gained will be used to represent all types of revenue models of the SLR platform. The SLR platform is operated by the SLR platform operator (asset broker), which brokers digital solutions, such as mobility services (assets) provided by software companies (asset providers), to counties (asset customers) and their citizens.
7) The second research project, *Machine Sharing Platform* (MSP), focuses on improving the production process of small and medium-sized enterprises through a platform that allows the sharing of machine capacities between manufacturers. The digital platform is operated by the MSP operator (asset broker), which brokers machine capacities, such as CNC milling machines (assets), between companies that have unused capacities of their machine tools (asset providers), and companies that have production bottlenecks and need these capacities for their own production processes (asset consumers).

4 Findings

This paper presents several key findings. First, it presents a final version of the taxonomy that includes all changes. Second, it outlines the changes that were identified during the development process of the revised taxonomy for revenue models of platform business models. Third, the paper provides an overview of several use cases that were analyzed using the taxonomy.

4.1 Taxonomy for Revenue Models of Platform Business Models

As discussed in the taxonomy building of Bartels et al. [7], we argue that a revenue model of a platform business model can only be described holistically if both the perspective of the asset broker as operator of the digital platform, and the perspective of the asset provider on the digital platform are reflected. Accordingly, the final taxonomy shown in Fig. 3 comprises 79 characteristics in 15 dimensions to take both perspectives into account. The taxonomy meets all ending conditions, and we claim that the taxonomy is complete.

A *revenue model type of the asset broker* (DB1) covers the revenue source and revenue stream through which the asset broker generates revenues. A *revenue stream of the asset broker* (DB2) describes how the asset broker generates revenues, i.e., the strategy the asset broker uses to monetize the revenue source through the platform. Access fees for platform participation, access fees for platform features, commission fees, a sales model of platform services, advertising fees for space, listing fees, or donations and sponsorships may be used to generate revenues. *The revenue source of the asset broker* (DB3) describes who is monetized by the asset broker, i.e., the actor through whom the asset broker generates the revenue stream. Here, asset consumers, asset providers, or third parties can be monetized by the asset broker. The *payment trigger of the platform price* (DB4) describes when payments recur for the asset broker, i.e., the point at which the revenue source is charged by the broker. Pay per platform access, pay per platform service use, pay per asset transaction, pay per asset listing, pay per user-related contact data are points where payments can be triggered for the asset broker. Additionally, revenue sources can have the flexibility to choose when to pay (pay whenever they want). The *payment frequency of the platform price* (DB5) describes how often payments recur for the asset broker, i.e., the frequency with which the revenue source is charged by the asset broker. Payments can be made on a one-time basis or on a recurring basis. The *price discovery of the platform price* (DB6) describes who sets the platform price, i.e., whether the platform price is set by the asset broker, by asset providers, asset consumers, or by negotiations. The *price mechanism of the platform price* (DB7) describes the influence of supply and demand on the platform price, i.e., whether the platform price is fixed or variable. The price of a platform can be fixed as either an absolute or percentage value, or it can be variable and negotiated, or have no constraints (pay what you want). If *price discrimination of the platform price* (DB8) exists, it can be described by different platform prices that are influenced by discriminatory factors, i.e., whether such factors affect the price to be paid on the platform. Platform price discrimination can take various forms, such as differentiating based on user type, user location, asset type, asset quantity, or through different platform tariffs, such as basic, pro, or premium tariffs.

A *revenue model type of the asset provider* (DP1) covers the revenue source and revenue stream by which the asset providers generate revenues. The *revenue stream of the asset provider* (DP2) describes how the asset providers generate revenues, i.e., the strategy the asset providers use to monetize the revenue source through the platform. The asset provider can generate revenue through the platform by selling, renting, charging a usage-based fee for the asset, or receiving donations and sponsorships. The *revenue source of the asset provider* (DP3) describes who is monetized by the asset providers, i.e., the actor through which asset providers generate their revenue stream. Asset consumers, the asset broker, or third parties can generate revenue for the asset providers. The *payment frequency of the asset price* (DP4) describes how often payments recur for asset providers, i.e., the frequency with which the revenue source is charged by the asset providers. Payments for an asset can be made either as a one-time payment at the time of asset purchase, with each asset subscription, with each use of the asset, or with each rental of the asset. Alternatively, payments can be left to the discretion of the revenue source via the pay whenever you want option. The *price discovery of the asset price* (DP5) describes who sets asset prices on the platform, i.e., whether asset prices are set by the asset broker, by asset providers, by asset consumers, or by negotiations. The *price mechanism of the asset price* (DP6) describes the influence of supply and demand on asset prices, i.e., whether asset prices on the platform are fixed or variable. The price of an asset can be a fixed and listed price, or it can be variable price and dependent on the current

Revenue model			Revenue model characteristics of a platform business model							
Revenue model of the asset broker	DB1	Revenue model type of the asset broker	Access model		Listing model	Advertising model	Commission model	Sales model	Donation and Sponsorship model	Other
	DB2	Revenue stream of the asset broker	Access fees for platform participation	Access fees for platform features	Listing fees on platform	Advertising fees for space	Commission fees	Sales model of platform services	Donations or sponsorships	Other
	DB3	Revenue source of the asset broker	Asset consumers		Asset providers			Third party		Other
	DB4	Payment trigger of the platform price	Pay per platform access		Pay per asset listing	Pay per user-related contact data	Pay per asset transaction	Pay per platform service use	Pay whenever you want	Other
	DB5	Payment frequency of the platform price	Pay once			Pay on a recurring basis				Other
	DB6	Price discovery of the platform price	Platform price set by asset broker		Platform price set by asset providers		Platform price set by asset consumers		Platform price set by negotiation	Other
	DB7	Price mechanism of the platform price	Absolute value		Percentage value		Variable (negotiated) value		Pay what you want	Other
	DB8	Price discrimination of the platform price	Type of asset	Type of user	Quantity of asset	Location of user	Different platform tariffs		No price discrimination	Other
Revenue model of the asset provider	DP1	Revenue model type of the asset provider	Sales model		Rental model		Pay per use model		Donation and sponsorship model	Other
	DP2	Revenue stream of the asset provider	Sales of assets		Rental fees for assets		Usage fees for assets		Donations or sponsorships	Other
	DP3	Revenue source of the asset provider	Asset consumers		Asset broker			Third party		Other
	DP4	Payment frequency of the asset price	Pay per asset subscription	Pay per asset use	Pay per rent		Pay once		Pay whenever you want	Other
	DP5	Price discovery of the asset price	Platform price set by asset broker		Platform price set by asset providers		Platform price set by asset consumers		Platform price set by negotiation	Other
	DP6	Price mechanism of the asset price	Fixed asset pricing				Variable asset pricing			Other
	DP7	Price discrimination of the asset price	Quantity of asset		Location of user		Type of user		No price discrimination	Other

Fig. 3. Revised taxonomy after the third iteration

demand. If *price discrimination of asset prices* (DP7) exists, it can be described in terms of different asset prices that are influenced by discriminatory factors on the platform. Asset price discrimination can take the form of asset quantity, user location, or user type.

4.2 Findings from the Empirical-to-Conceptual Taxonomy Development Process

The revisions made to the taxonomy in Fig. 1, are reflected in Table 1 with 34 total changes, based on two empirical-to-conceptual iterations. Table 1 provides a categorized documentation of all changes made, including a change number (N°), and the iteration in which the change occurred (2nd or 3rd iteration). Additionally, it identifies the platform business model that prompted the change, such as Tyre24.

Table 1. Documented changes of the taxonomy development process

N°	Iter.	Causer	At	Type	Impact	Change
1	2nd	Tyre24	DB1	Duplicate	Moderate	'Pay-per-use model' is replaced by 'pay per' in DB4
2	2nd	nebenan	DB1	Extend	Moderate	'Donation model' is extended to 'donation and sponsorship model'
3	2nd	Tyre24	DB2	Split	Moderate	'Access fees' is split into 'participation in a platform' and 'access to platform services'
4	2nd	Tyre24	DB2	Merge	Moderate	'Commission fees on platform transactions' and 'commission fees on usage' is merged into 'commission fees'
5	2nd	nebenan	DB2	Extend	Moderate	'Donations' is extended to 'donations and sponsorships'
6	2nd	n/a	DB4	Replace	Minor	'One-time' is replaced by 'pay once'
7	2nd	Tyre24	DB4	Replace	Minor	'Subscription-based frequency' is replaced by 'pay on a recurring basis'
8	2nd	My Hammer	DB4	Split	Moderate	'Usage-based frequency' is split into 'pay per platform access', 'pay per asset listing', 'pay per user-related contact data', 'pay per asset transaction' and 'pay per platform service use'
9	2nd	empto	DB6	Split	Moderate	'Fixed platform pricing' is split into 'absolute value' and 'percentage value'
10	2nd	Vinted	DB6	Replace	Minor	'Variable platform prices' is replaced by 'variable (negotiated) value'
11	2nd	My Hammer	DB7	Split	Moderate	'Feature-based price discrimination' is split into 'type of asset', 'type of user', 'location of user' and 'different platform tariffs'
12	2nd	Vinted	DB7	Replace	Minor	'Quantity-based price discrimination' is replaced by 'quantity of asset'
13	2nd	n/a	DB7	Replace	Minor	'Location-based price discrimination' is replaced by 'location of user'
14	2nd	empto	DB7	Extend	Moderate	'No price discrimination' is added
15	2nd	nebenan	DP1	Extend	Moderate	'Donation and sponsorship model' is added
16	2nd	nebenan	DP2	Extend	Moderate	'Donations and sponsorships' is added
17	2nd	n/a	DP4	Replace	Minor	'Subscription-based frequency' is replaced by 'pay per asset subscription'
18	2nd	n/a	DP4	Split	Moderate	'Usage-based frequency' is split into 'pay per asset use', 'pay per rent' and 'pay once'
19	2nd	nebenan	DP4	Extend	Moderate	'Pay whenever you want' is added
20	2nd	Vinted	DP7	Replace	Minor	'Feature-based price discrimination' is replaced by 'type of user'
21	2nd	n/a	DP7	Replace	Minor	'Quantity-based price discrimination' is replaced by 'quantity of asset'
22	2nd	n/a	DP7	Replace	Minor	'Location-based price discrimination' is replaced by 'location of user'
23	2nd	Tyre24	DB7	Extend	Moderate	'No price discrimination' is added
24	3rd	SLR	n/a	Extend	Major	New dimension 'payment trigger' is added
25	3rd	SLR	DB4	Replace	Moderate	'Pay per platform subscription' is replaced by 'pay per platform access' and is added to new dimension in N° 24
26	3rd	SLR	DB4	Replace	Moderate	'Pay per platform service use' is added to new dimension in N° 24
27	3rd	SLR	DB4	Replace	Moderate	'Pay per asset transaction' is added to new dimension in N° 24
28	3rd	SLR	DB4	Replace	Moderate	'Pay per asset listing' is added to new dimension in N° 24
29	3rd	SLR	DB4	Replace	Moderate	'Pay per user-related contact data' is added to new dimension in N° 24
30	3rd	SLR	DB4	Replace	Moderate	'Pay what you want' is replaced by 'pay whenever you want' and is added to new dimension in N° 24
31	3rd	SLR	DB4	Extend	Moderate	'Other' is added
32	3rd	SLR	DB4	Replace	Minor	Description of 'pay once' is replaced
33	3rd	SLR	DB4	Replace	Minor	'Pay per platform subscription' is replaced by 'Pay on a recurring basis'
34	3rd	SLR	DB7	Extend	Minor	'Pay what you want' is added

The dimension where the change occurred is noted, ranging from DB1 to DP7. The specific type of modification, whether it is a duplication or an extension, is stated as well. The table further categorizes the level of impact each change had on the taxonomy, classifying it as minor, moderate, or major. Finally, a description of each change is provided to give more context and details. Changes that were not triggered by a specific platform as causer but emerged during the taxonomy development process are marked as not applicable (n/a). As outlined in Sect. 3, two empirical-to-conceptual iterations were carried out that led to a holistic improvement of the taxonomy. In the third iteration, when we applied the taxonomy to real-life use cases, we assumed that there would be no more significant changes. However, we identified a completely new dimension, ‘payment trigger’, which was of great value, but surprisingly occurred surprisingly at a late stage in the development process.

The development process was considered complete as no further changes were identified in the application of the Machine Sharing Platform as the second use case during the third iteration, resulting in a stable state of the taxonomy. The types of changes were classified into five categories: duplicate, extend, split, merge, and replace, with descriptions provided for each. Of the 34 changes made, 68% were identified in the first iteration of the empirical-to-conceptual development phase. However, the SLR platform observed in the last iteration caused the most changes to the taxonomy, accounting for 32% of all changes. This is because a new dimension, ‘payment trigger’, was identified. The revenue model of the SLR platform employs a listing model, where each digital solution listed on the platform incurs a one-time fee and a recurring fee for the asset provider. Therefore, the payment trigger ‘pay per asset listing’ should be distinguished from the payment frequency dimension and its characteristics ‘pay once’ or ‘pay on a recurring basis’, as combining them as one dimension would not differentiate the listing model of the SLR platform sufficiently. This change had a significant impact on the taxonomy. Other dimensions, such as ‘revenue source’ (DB3) and ‘price discovery’ (DB6), were not changed during the iterations. Among the types of changes, ‘replace’ (50%) and ‘extend’ (29%) were the most frequent. The majority of changes had a moderate impact on the taxonomy, with nearly 62% involving only name or description changes to individual characteristics. In summary, the two empirical-to-conceptual iterations led to a more comprehensive and robust taxonomy than the initial version in Fig. 1. For a more detailed documentation of the changes made over each iteration, see [9].

4.3 Presentation of the Analyzed Platform Revenue Models

The taxonomy analysis and examination of the revenue models of the seven platform business models led to the identification of 26 distinct revenue model types for the asset brokers, which are presented in Table 2. For each platform business model the summary is based on three aggregated dimensions: (1) *Who is monetized?*, (2) *how is it monetized?*, and (3) *how much is monetized?*. As shown in Table 2, the number of revenue model types varies among the seven platform

Table 2. Analyzed revenue model types for each platform business model

N°	Platform	Who pays?	How is it monetized?	How much is monetized?
1	***Tyre24***	Consumers	Access fees to participate	€29 or €69 monthly
2	***Tyre24***	Consumers	Commission fees	3.9% or 1.9% per transaction
3	***Tyre24***	Consumers	Access fees to service	Free or €99 monthly
4	***Tyre24***	Providers	Access fees to service	Free or €99 monthly
5	***Tyre24***	Providers	Commission fees	Free or 0.9% per transaction
6	***empto***	Consumers	Commission fees	4% per transaction
7	***empto***	Consumers	Commission fees	4% per transaction
8	***My Hammer***	Providers	Commission fees	€1-€89 per user contact
9	***Vinted***	Consumers	Commission fees	5% per transaction
10	***Vinted***	Consumers	Protection service	€0.7 per transaction
11	***Vinted***	Consumers	Verification service	€25 per item
12	***Vinted***	Providers	Item visibility service	On demand
13	***Vinted***	Providers	Best matches service	€6.95 per item per week
14	***Vinted***	Third party	Fees for advertising space	On demand
15	***nebenan***	Consumers	Donations for platform	Pay what you want
16	***nebenan***	Providers (business)	Access fees to participate	€12, €19, or €49 monthly
17	***nebenan***	Providers (organizations)	Access fees to participate	€10, €18, or €50 monthly
18	***nebenan***	Third party	Sponsorship with platform	On demand
19	***nebenan***	Third party	Fees for advertising space	On demand
20	***SLR***	Consumers	Access fees to participate	€500 one-time
21	***SLR***	Consumers	Access fees to participate	€140 monthly
22	***SLR***	Providers	Listing fees for asset	€1.000 one-time
23	***SLR***	Providers	Listing fees for asset	€250 monthly
24	***MSP***	Consumers	Access fees to participate	€5250 one-time
25	***MSP***	Providers	Access fees to participate	€5250 one-time
26	***MSP***	Providers	Commission fees	23% per transaction

business models. The Vinted platform has the largest number of revenue model types with six, followed by Tyre24 and nebenan.de with five each. On the other hand, the MyHammer platform monetizes its entire platform business model with a single commission model and shows the variety in revenue model types. Overall, it can be said that about 46% of the 26 examined revenue model types use asset consumers as their revenue source, 42% use asset providers, and only 12% use third parties. The most common revenue streams are access models with 35% and commission models with 27%. The analysis of the revenue models of the asset providers showed that they often rely on selling their own assets, with sales models representing 75% of their revenue streams on the platform. 88% of the revenue sources for asset providers are from asset consumers and 13% from asset brokers. Further information is available in the research data [9]. Tyre24 offers basic and premium access models for platform participation at €29 and €69 per month, each with different features. The platform generates revenue by monetizing car repair shops as asset consumers who buy car parts and accessories. Commission models apply, with 3.9% for basic access and 1.9% for premium access.

Suppliers of car repair items are also monetized, with transaction fees of 0.9% depending on their commission group. The revenue model for asset providers is based solely on sales of their car parts on the platform. The empto platform charges a 4% commission fee per transaction for both waste producing companies and waste disposers. The platform's revenue model for asset providers is solely based on sales of their disposal services on the platform, without any additional revenue streams. The revenue model of the MyHammer platform is based on a commission model, where skilled trade businesses that act as asset providers are charged a fee for each confirmed contact with a householder. Commission fees vary based on the type and scope of the trade service, ranging from €1 to €89 per contact confirmation. The revenue model for asset providers is solely based on sales of their trade services on the platform, without any additional revenue streams. Vinted generates revenue through a commission model and platform service fees for buyer protection and item verification. The platform charges buyers a 5% commission fee per transaction, a mandatory €0.70 fee for buyer protection, and an optional item verification service fee is €25 per item. Sellers can purchase 'bumps' to increase the visibility of their clothing items. Third parties are monetized through an advertisement model on the platform. The revenue model for asset providers is based solely on sales of their clothing items. nebenan.de charges local businesses and non-profit organizations an access fee ranging from €10 to €50 per month for publishing posts. nebenan.de generates revenue through donations, voluntary contributions, and partnerships with cities and municipalities. Local organizations can earn donations, while the platform increases the visibility of local shop products without selling directly. The SLR platform generates revenue through an access model and a listing model. Counties pay a one-time fee of €500 and a monthly fee of €140 to participate in the platform, while software companies pay a one-time fee of €1.000 and a monthly fee of €250 for each solution listed on the SLR platform. The revenue model for asset providers is based solely on sales of their digital solutions. The Machine Sharing Platform charges a one-time participation fee of €5250 for companies to access the platform and monetizes asset providers through a 23% commission on each transaction. The revenue model for asset providers is based solely on sales of their machine capacities on the platform.

5 Discussion and Conclusion

This study provides three main contributions: (1) a detailed and extensive taxonomy of revenue models of platform business models; (2) evidence that platform business models can adopt various revenue model types that can be creatively combined to develop innovative monetizing strategies; and (3) an analysis of seven platform business models resulting in the identification of 26 distinct revenue model types. The research question of how to classify revenue models of platform business models is answered with the applied taxonomy, which comprises 15 dimensions and 79 characteristics. We consider the use of the empirical-to-conceptual iteration cycles in this research successful, as we found 34 changes from the initial taxonomy and created a revised taxonomy. 68% of all changes were found in the first iteration cycle, while 32% were found in the second. Five dimensions with their associated characteristics were not changed, whereas the remaining ten dimensions underwent changes ranging from minor character changes to major dimensional adjustments. Based on the data we analyzed [9], we found that the most common revenue model types of the 26 revenue model types we analyzed fall into the category of access model (35%) and commission model (27%). The cluster analysis conducted by Täuscher and Laudien [22] shows that 72% of the 100 examined marketplaces generate revenue through commission fees. The authors applied their taxonomy to each platform business model once. In contrast, we applied our taxonomy multiple times for each platform business model, as we identified various revenue model types for a single platform (e.g. Tyre24). Five of the seven platform business models we examined (71%) operate a commission model (Tyre24, empto, MyHammer, Vinted, and MSP), which aligns with the finding of Täuscher and Laudien (72%). However, when analyzing all 26 identified revenue model types across the seven examined platform business models, it becomes apparent that commission models account for only 27% of the total. This observation suggests that platforms such as Tyre24 employ multiple revenue model types simultaneously. This leads to the conclusion that platform revenue models often involve a combination of revenue model types. As a result, describing platform revenue models using a single revenue model type may not accurately capture the diverse approaches used to generate revenues.

Limitations. We made an effort to ensure that the development of the presented taxonomy was as transparent as possible and are documented in the research data. However, there are limitations to our study that need to be addressed. The use cases we selected represent only a small portion of existing platform business models, and are focused on platforms that are mainly operated in Germany, so that the results may have regional constraints. Other or additional use cases may lead to changes in the taxonomy that were not captured in this paper. Although the taxonomy was developed with great care through both conceptual-to-empirical and empirical-to-conceptual iteration cycles and claims to be complete, we cannot guarantee this. It is important to note that the taxonomy is only stable until further iterations reveal new potential dimensions and characteristics. We expect the taxonomy to be stable, as no further changes were identified during the last case study.

Future Work. On the proposed taxonomy should explore various revenue models of platform businesses in real-world settings to evaluate the validity of the proposed taxonomy. The updated taxonomy guidance proposed by Kundisch et al. [16], extends the approach of Nickerson et al. [18], providing taxonomy designers with a method to evaluate their developed taxonomies. Based on this, our next step is to integrate our taxonomy into the business model design process and assess its applicability in supporting practitioners define appropriate platform revenue models. The goal of this research is to make the taxonomy available as a design tool for practitioners to systematically create revenue models, as suggested by Bartels and Gordijn [6]. We also aim to understand the dynamic changes in platform business models, from the initial phase of a platform with a small user base to a stage with a critical mass of asset providers and consumers, potentially enabling reinforcing network effects. We posit that different development stages of a platform could influence the design choices of a platform operator regarding an appropriate revenue model. Such a model must incentivize providers and consumers (for example, through sign-up discounts) while also capturing value through monetization mechanisms. Furthermore, future studies should also explore the correlation of different types of revenue models for specific platform business models. For instance, a donation-based revenue model, like the one implemented by nebenan.de, may have different enablers or barriers compared to a transaction-based revenue model, such as the one employed by the empto platform. We believe this complex business model dynamics and evolvement requires further exploration. A well-constructed taxonomy can contribute to theory building by representing forms of descriptive knowledge [18]. In this regard, our taxonomy can serve as an instrument for extracting unidentified knowledge about platform revenue model strategies. This newfound knowledge could be ensured in the form of platform revenue model archetypes, similar to the approach taken by Bergman et al. [11] in extracting business model archetypes for data marketplaces within the automotive industry. Our aim is to identify platform revenue model archetypes as design patterns that reflect proven design knowledge. Therefore, we want to create a design tool that can be used by practitioners in various settings such as interactive workshops, thereby enhancing the accessibility and practicality of this knowledge.

Acknowledgments. We thank Sonnhild Namingha for linguistic revision. The research was conducted as part of the Smarte.Land.Regionen project (2818SL001).

References

1. Homepage of empto. https://www.empto.de/. Accessed 30 Apr 2023
2. Homepage of MyHammer. https://www.my-hammer.de/. Accessed 30 Apr 2023
3. Homepage of nebenan.de. https://nebenan.de/. Accessed 30 Apr 2023
4. Homepage of tyre24. https://tyre24.alzura.com/. Accessed 30 Apr 2023
5. Homepage of Vinted. https://www.vinted.de/. Accessed 30 Apr 2023
6. Bartels, N., Gordijn, J.: A business model construction kit for platform business models - research preview. In: Gervasi, V., Vogelsang, A. (eds.) REFSQ 2022. LNCS, vol. 13216, pp. 175–182. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-98464-9_14
7. Bartels, N., Koch, M., Gordijn, J.: Developing a taxonomy for revenue models of platform business models. In: 36th Bled eConference Digital Economy and Society - the Balancing Act for Digital Innovation in Times of Instability: Conference Proceedings. University of Maribor Press (2023). https://doi.org/10.18690/um.fov.4.2023.1
8. Bartels, N., Koch, M., Gordijn, J.: Reserach data taxonomy for revenue models of platform business models. Zenodo (2023). https://doi.org/10.5281/zenodo.7635726
9. Bartels, N., Koch, M., Schmitt, A., Gordijn, J.: Reserach data taxonomy for platform revenue models. Zenodo (2023). https://doi.org/10.5281/zenodo.7965631
10. Bartels, N., Schmitt, A.: Developing network effects for digital platforms in two-sided markets – the NfX construction guide. Digit. Bus. **2**(2), 100044 (2022). https://doi.org/10.1016/j.digbus.2022.100044

11. Bergman, R., Abbas, A.E., Jung, S., Werker, C., de Reuver, M.: Business model archetypes for data marketplaces in the automotive industry. Electron. Mark. **32**(2), 747–765 (2022). https://doi.org/10.1007/s12525-022-00547-x
12. Freichel, C., Fieger, J., Winkelmann, A.: Developing a taxonomy for digital platforms – a conceptual approach. In: Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences (2021). https://doi.org/10.24251/hicss.2021.701
13. Gordijn, J., Wieringa, R.: E3value user guide - designing your ecosystem in a digital world. The Value Engineers, 1st edn. (2021)
14. Koch, M., Krohmer, D., Naab, M., Rost, D., Trapp, M.: A matter of definition: criteria for digital ecosystems. Digit. Bus. **2**(2), 100027 (2022). https://doi.org/10.1016/j.digbus.2022.100027
15. Koch, M., et al.: Digitale Ökosysteme in Deutschland. Fraunhofer Publica (2023). https://doi.org/10.24406/publica-1011
16. Kundisch, D., et al.: An update for taxonomy designers. Bus. Inf. Syst. Eng. **64**(4), 421–439 (2021). https://doi.org/10.1007/s12599-021-00723-x
17. Moore, J.F.: The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems. Wiley, Hoboken (1996)
18. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. Eur. J. Inf. Syst. **22**(3), 336–359 (2013). https://doi.org/10.1057/ejis.2012.26
19. Parker, G.G., Van Alstyne, M.W., Choudary, S.P.: Platform Revolution: How Networked Markets are Transforming the Economy and How to Make them Work for You. WW Norton & Company, New York (2016)
20. Springer, V., Petrik, D.: Towards a taxonomy of impact factors for digital platform pricing. In: Gregory, P., Kruchten, P. (eds.) XP 2021. LNBIP, vol. 426, pp. 115–124. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88583-0_11
21. Staub, N., Haki, K., Aier, S., Winter, R.: Taxonomy of digital platforms: a business model perspective. In: Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences (2021). https://doi.org/10.24251/hicss.2021.744
22. Täuscher, K., Laudien, S.M.: Understanding platform business models: a mixed methods study of marketplaces. Eur. Manag. J. **36**(3), 319–329 (2017). https://doi.org/10.1016/j.emj.2017.06.005
23. Teece, D.J.: Business models, business strategy and innovation. Long Range Plann. **43**(2), 172–194 (2010). https://doi.org/10.1016/j.lrp.2009.07.003
24. van de Ven, M., Abbas, A.E., Kwee, Z., de Reuver, M.: Creating a taxonomy of business models for data marketplaces. In: 34th Bled eConference Digital Support from Crisis to Progressive Change: Conference Proceedings. University of Maribor Press (2021). https://doi.org/10.18690/978-961-286-485-9.23
25. Weking, J., Hein, A., Böhm, M., Krcmar, H.: A hierarchical taxonomy of business model patterns. Electron. Mark. **30**(3), 447–468 (2018). https://doi.org/10.1007/s12525-018-0322-5

Conceptual Modeling in Support of Economic and Regulatory Viability Assessment - A Reality Check on the Example of Developing an Energy Community

Sybren de Kinderen[1](✉), Qin Ma[2], Monika Kaczmarek-Heß[3], and Rik Eshuis[1]

[1] Information Systems Group IE&IS, Eindhoven University of Technology, Eindhoven, The Netherlands
{s.d.kinderen,r.eshuis}@tue.nl

[2] University of Luxembourg, 2 Avenue de l'Université, 4365 Esch-sur-Alzette, Luxembourg
qin.ma@uni.lu

[3] University of Duisburg-Essen, Universitätstrasse 9, 45141 Essen, Germany
monika.kaczmarek-hess@uni-due.de

Abstract. This paper offers an assessment of the extent to which conceptual modeling can be used for a conjoint assessment of regulatory and economic viability of new projects in the electricity sector, with a particular focus on developing energy communities. To this end, we establish a set of challenges resulting out of a confrontation of, on the one hand, the observed relevance of conjointly assessing the regulatory and economic viability for electricity sector projects, and on the other hand, the fact that no dedicated efforts exist which explicitly target such a conjoint assessment. Then, using a realistic scenario we show how two selected conceptual modeling languages can be used for a conjoint assessment: Legal GRL, for regulatory viability, and e^3*value*, for economic viability. Finally, we discuss lessons learned from our experience, among others, a need for a taxonomy for energy sector specific regulation and the use of value network patterns.

Keywords: Regulatory and Economic Viability · Conceptual Modeling · Energy Community

1 Introduction

With the increasingly decentralized production and consumption of electricity [10,32], whereby citizens are increasingly involved in both electricity production and trading [32], as well as the increasing use of renewables [23], the electricity sector is undergoing remarkable changes. This is also reflected in a host of new electricity projects, such as the increasing emergence of energy communities in their various incarnations [21,32].

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 206–222, 2024.
https://doi.org/10.1007/978-3-031-46587-1_12

The development of new electricity sector projects requires an assessment of their technical, strategic, and business model viability [10,18,29]. Nevertheless, of equal importance is an assessment of the regional context in which the project is to be carried out [16,30], whereby especially *regulation* is considered to be of high relevance [3,23]. In particular, the rights and responsibilities associated with regulation have a considerable influence on early project development in terms of influencing the role that actors play [9, p. 10], the distribution of costs and benefits [30, p. 43][9, p. 10], or on the extent to which benefits can be realized [16, p. 12]. As such, regulation has a non-trivial effect on early project development, often going beyond a binary check on whether something is allowed (or not).

To clarify the relevance of regulation during the early development of electricity sector projects, as a particular instance consider the impact of regulation on the development of energy communities. Although the notion of an energy community is hard to pin down specifically [32], generally speaking, an energy community can be said to refer to a group of citizens with shared common objectives relating to the production, distribution, and supply of energy [26]. Here especially, the citizen is important, in the sense that citizens can actively partake in electricity generation and selling [22], as well as a sense of community [32], which can indeed refer to a local community, although an energy community is not necessarily tied to a specific place [26].

When it comes to developing an energy community in a particular region, experience shows that explicitly considering regulation is very important [5,21]. This is particularly visible for energy communities in different European Union member states, whereby different member states showcase a high variability in terms of the regulatory transposition of EU-level regulation on energy communities [21]. To illustrate this, consider the energy community Schoonschip. Schoonschip, also the running scenario for this paper, is an energy community of houseboats located in the north of Amsterdam, the Netherlands [5,14,25]. It is composed of 46 households generating their own electricity with photovoltaic solar panels. All homes are connected to the local Schoonschip smart grid, which has a single connection to the public grid. During project development Schoonschip and its members had the ambition to be self-sufficient in their electricity supply, to operate their own grid, to sell part of the surplus electricity from the community back to the grid, and to contribute to the sustainable energy transition [5]. Nevertheless, as we discuss in Sect. 3, due to regulatory concerns Schoonschip had to make several adjustments to its initial ambitions. One example concerns the regulation dictating the right of consumers to freely choose electricity suppliers. Since this regulation allows Schoonschip community members to pick an external electricity supplier, which is moreover not guaranteed to focus on green electricity, both the ambitions of being self-sufficient and contributing to the sustainable energy transition, are affected. In this example, regulation not only influences the distribution of costs and benefits (e.g., benefits partly flow to an external supplier), but also influences the extent to which benefits can be realized (e.g., members are not guaranteed to source green electricity).

Given the influential role of regulation, we call for an instrument to allow for identifying relevant regulations and analyzing regulatory viability conjointly with early-stage electricity sector project development. Conceptual modeling, here referring to "the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication" [28] has already been used to support project development in the electricity sector. Various experiences [10,18,31] show that conceptual modeling can help early project development in terms of (1) gaining conceptual clarity, by offering a focused set of concepts, (2) fostering shared understanding among stakeholders, and (3) facilitating semi-automated reasoning, in terms of cash flow analysis or goal satisfaction analysis. Whereas conceptual modeling has been already applied to support regulatory analysis, for an overview see [2], as well as economic analysis, e.g., [1,7,11], to the best of our knowledge no dedicated approaches exist that support a conjoint assessment of economic and regulatory viability.

Considering it, the objective of this paper, being part of a larger design-science research project focusing on support of conceptual modeling for the electricity sector, is by relying on a realistic scenario, to highlight the role that conceptual modeling can play in a conjoint assessment of regulatory and economic viability of new projects in the electricity sector. Note here that we consciously focus on assessing economic viability as part of early project development. We do so for scoping purposes, as this focus allows us to capitalize on a well-established set of modeling techniques, which show nicely the influence of regulation.

In our previous work [19], we established the relevance of the conjoint analysis of assessing business development concerns and regulatory concerns. This paper continues our effort by contributing (1) a set of challenges to which conceptual modeling should respond, (2) showing how, for our purposes, we can conjointly use two existing modeling languages dedicated to, respectively, regulatory analysis (Legal GRL) and value exchange analysis (e^3*value*). We do so based on a particular, Schoonschip, case and (3) how an analysis can be made for regulation specific to the electricity sector. To this end we borrow inspiration from both, the development of regulatory models specific to energy communities, as well the use of the modeling method for selecting relevant regulations. Thus, we address in this paper the intersection of problem investigation and treatment design cycles as defined by [36].

Note that with these contributions, this paper emphasizes experiences from using conceptual modeling for a particular case study, to gather input for a future modeling method. As such, presenting a fully-fledged modeling method is not its aim yet (though various elements of a future method, like the followed steps, and used taxonomy, are already visible). Our work stands in a tradition of the conjoint use of modeling languages, such as [13] for business process operationalization of a value network, or [34] for IT risk management. Nevertheless, we target a novel combination of modeling languages, in light of a practical problem that has not been sufficiently tackled before (as per Sect. 2).

This paper is structured as follows. In Sect. 2 we discuss how regulation and business development are intertwined using the introduced Schoonschip energy community as a running example. Also, we derive a set of challenges to which conceptual modeling can contribute. Thereafter, in Sect. 3 we discuss how modeling can support a conjoint economic and regulatory analysis. Here, we employ again Schoonschip as an illustrative scenario. Finally, in Sect. 4 we reflect on our use of modeling and discuss the next steps.

2 Regulation and Business Development: Challenges

The relevance of regulation in business development methods for the electricity sector generally [9,16,30], and for developing energy communities specifically [3,21] has already been observed in the literature. Indeed, the rights and duties implied by regulation *actively* shape business development considerations, instead of merely posing a binary question of an energy community being compliant (or not). Such active shaping expresses itself, among others, in determining the following aspects of business development: (1) the involved actors and the role they play [9, p. 10]; (2) how costs and benefits are distributed across actors [30, p. 43][9, p. 10]; and (3) the extent to which regional regulatory conditions are suitable for realising the ideal type benefits as originally envisioned for an electricity sector project [16, p. 30].

The described Schoonschip scenario also aligns with these observations. For example, regarding the first aspect, i.e., regulation shaping the involved actors and the role they play, under the Dutch regulation Schoonschip must let a Distributed System Operator (DSO) carry out management tasks of the local grid such as managing the transactions for energy sharing and selling, which would ideally be managed by community members (see [20, p. 163] and the Rescoop.eu transposition tracker[1]). Regarding the second aspect, i.e., regulation shaping how costs and benefits are distributed across actors, Schoonschip expects its members to exchange energy among themselves and to set their costs and tariffs. However, the right of members to freely choose an energy supplier, as allowed by the regulation, creates a lot of uncertainty, as part of the payment for the electricity may flow outside the energy community [5]. Finally, regarding the third aspect, i.e., regulation shaping the extent to which the original ideal type benefits can be realized with regional regulatory conditions, in the Netherlands energy communities are not subject to favorable network charges nor special tax exemptions [20, p. 164], which has a significant effect on the financial feasibility of the Schoonschip project.

Nevertheless, as we observed in our previous work [19], *how* regulation influences business development for electricity sector projects remains an open issue. This is echoed by authors of [30], who reviewed cost-benefit analysis methods for electricity sector projects and pointed out that despite its relevance, regulation is insufficiently considered, and also by the authors of [27], who call for making regulatory analysis a first-class citizen during early project development. We formulate the observation as

[1] https://www.rescoop.eu/transposition-tracker-support-schemes. The transposition tracker, initiated by a federation of European energy collectives, reviews the extent to which EU member states have transposed EU-level directives on energy communities into national regulation.

Challenge 1. *There is a need to systematically account for the non-trivial role of regulation during business development of electricity sector projects in general and of energy communities in particular, from the perspectives of how regulation influences: (a) the changing role of actors, (b) the distribution of costs and benefits, (c) the realization of ideal type benefits.*

Furthermore, with our focus on energy communities, it is important to explicitly account for the particularities of energy community regulation. Given a regulatory context, one shall first review regulatory definitions of energy communities. For example, in the Netherlands, the two definitions of a Citizen Energy Community (Directive (EU) 2018/2001) and a Renewable Energy Community (Directive (EU) 2019/944) are merged during the transposition, giving rise to a unified definition of energy communities. Secondly, one shall also review enabling legal frameworks and support schemes in place, which foster households to jointly act as a renewable energy community by e.g., allowing for peer-to-peer electricity exchange (cf. [15,20], Rescoop.eu).

As a baseline for systematically analysing energy community regulation (i.e., regulatory definitions, legal frameworks, and support schemes), one can adopt taxonomies of regulation relevant to innovation in the electricity sector, such as [3,4,33]. These taxonomies provide a useful point of departure by characterizing a given energy community in a regulatory context according to among others, the role of electricity sector actors, geography, allowed exemptions, and incentives [3]. However, existing taxonomies largely exist to enable comparison of regulation among (mostly) EU member states. To the best of our knowledge, they are not yet part of a more comprehensive instrument that considers regulation as part of business model development, like what we strive for. Thus, we formulate

Challenge 2. *There is a need to establish taxonomies accounting for the (regional) regulatory particularities of the electricity sector and energy communities, to enable a conjoint consideration of business development and regulatory viability.*

Finally, partly specific to energy communities, the users' engagement seems to be crucial [3,18,23]. To foster engagement, efficient communication, as well as an understanding of an initiative and its goals, are necessary, see, e.g., [32]. Therefore, we formulate

Challenge 3. *There is a need to account for social aspects of energy communities, among others the engagement of stakeholders.*

Considering these challenges, we argue that an instrument is needed that would help deal with the complexity and variability of regulation and its impact on the configuration of energy communities (Challenge 1 and 2), but at the same time enable effective communication among involved stakeholders to foster their engagement (Challenge 3). Conceptual modeling is known for its advantage in fostering communication and controlling complexity [34]. To understand better the role conceptual modeling can play in the design of energy communities, we explore in the context of Schoonschip how selected modeling languages used in tandem can support the conjoint analysis of interest in the next section.

3 Reality Check

As a point of departure, we follow the process depicted in Fig. 1. In terms of phases, this process takes inspiration from both (a) the cost-benefit analysis methods for smart grid projects generally [9,16,30], and (b) practical experiences in developing energy communities specifically [24]. For a given project idea, both (a) and (b) suggest firstly *identifying boundary conditions*, of which the identification of regional regulation is an important constituent [16], and secondly, *developing a business case* to identify among others, the customers, the value proposition, actors involved, and what the actors get out of project participation. Subsequent phases then vary between (a) and (b), whereby experiences from energy community development, e.g., [24], propose to proceed with *community building*, because the commitment of key stakeholders is critical to the success of the project, but not always straightforward to achieve, whereas cost-benefit analysis methods, e.g., [9] propose to continue with a sensitivity analysis step to tease out the effect of variation of key project variables on project variability. In this paper, we focus on the first two phases of the process for which the influence of regulation seems to be the most pertinent.

In particular, we start by identifying regional regulatory conditions (Step 1) to systematically analyze the regulatory environment of the project. Then, during the business case development phase, we first design a regulation-agnostic value network (Step 2), with a focus on showing the network of actors and performing per-actor cash flow calculations to gauge the project's economic viability as is typical during early project development [10].

Thereafter, taking the constituencies of the modeled value network as a point of departure, during "regulatory analysis of the value network" (Step 3), we analyze how regulations modeled in Step 1 influence the regulation-agnostic value network, e.g., in terms of reconsidering the role of actors, re-distribution of costs and benefits, and the extent to which benefits as modeled in the value network can indeed be realized under a given regulation (cf. Challenge 1 & 2). Based on the analysis, a feedback loop from Step 3 to Step 2 enables processing the found regulatory impacts on the value network (cf. Challenge 1 & 2).

Please note that the choice for first designing the value network, and only thereafter confronting it with regulations is motivated by, firstly, the selection of relevant regulations simply requiring that the project idea is first clarified, e.g., in terms of key activities and actors involved. Secondly, starting with an ideal type value network and only then confronting it with regulations is in line with the idea of early project development, to foreground first idea exploration prior to a (regulatory) reality check.

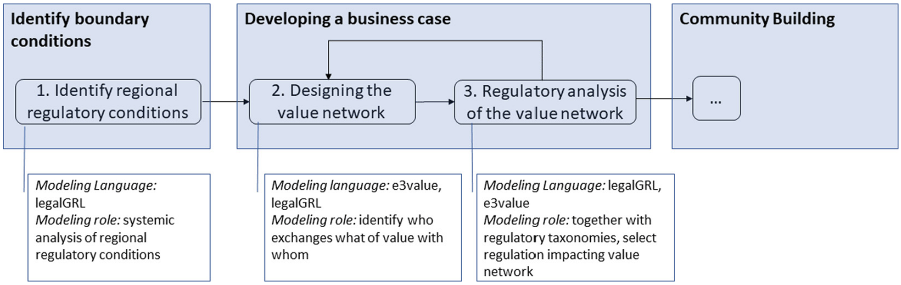


Fig. 1. Main steps followed, and the role of modeling therein

3.1 Step 1: Identify Regional Regulatory Conditions

To assess the regulatory viability of the Schoonschip energy community initiative, we first identify applicable regulations. According to [35], the Schoonschip energy community is ruled by a hierarchical regulatory environment, including EU directives (e.g., Directive 2018/2001 on the promotion of renewable energy sources and Directive 2019/944 on common rules for the internal electricity market), national legislation of the Netherlands (e.g., Environmental Tax Act), and regulatory sandboxes temporarily adopted by the Dutch government (e.g., the Experimentation Decree). Texts extracted from these regulations are then formalized in terms of conceptual models. For regulatory modeling, several approaches exist (for an overview, see [2]). We focus on Legal GRL, a conceptual modeling language extending the Goal-oriented Requirements Language (GRL) with capabilities to capture legal modalities (i.e., obligation and permission). We do so since Legal GRL offers strong software tool support[2], and it has a strong basis in the seminal work of Hohfeld from the juridical literature [12].

For regulatory modeling, Legal GRL offers the following Hohfeldian concepts: Permission (matching to the Hohlfeldian Duty-claim notion), Obligation (matching to Privilege-Noclaim), and Precondition. Also [8] offers guidance to translate regulatory text into legal GRL models. Nevertheless, due to a lack of space and to focus on the conjoint regulatory and value network analysis, we will mostly focus on the resulting models, less so on the regulatory text-to-model translation.

Illustration: Figure 2 presents the Legal GRL model capturing an excerpt of regulations applicable to Schoonschip. We zoom into three extracts from the text of these regulations to illustrate how to model them in Legal GRL.

Firstly, Article 4 of EU Directive 2019/944 (Free choice of supplier) states: "*...all customers are free to purchase electricity from the supplier of their choice...*". This statement applies to Schoonschip, as customers consuming electricity also produce electricity through the solar panels on their rooftops, hence are prosumers. We model it as a «Permission» goal owned by the actor Prosumer in the Legal GRL model.

[2] https://github.com/JUCMNAV.

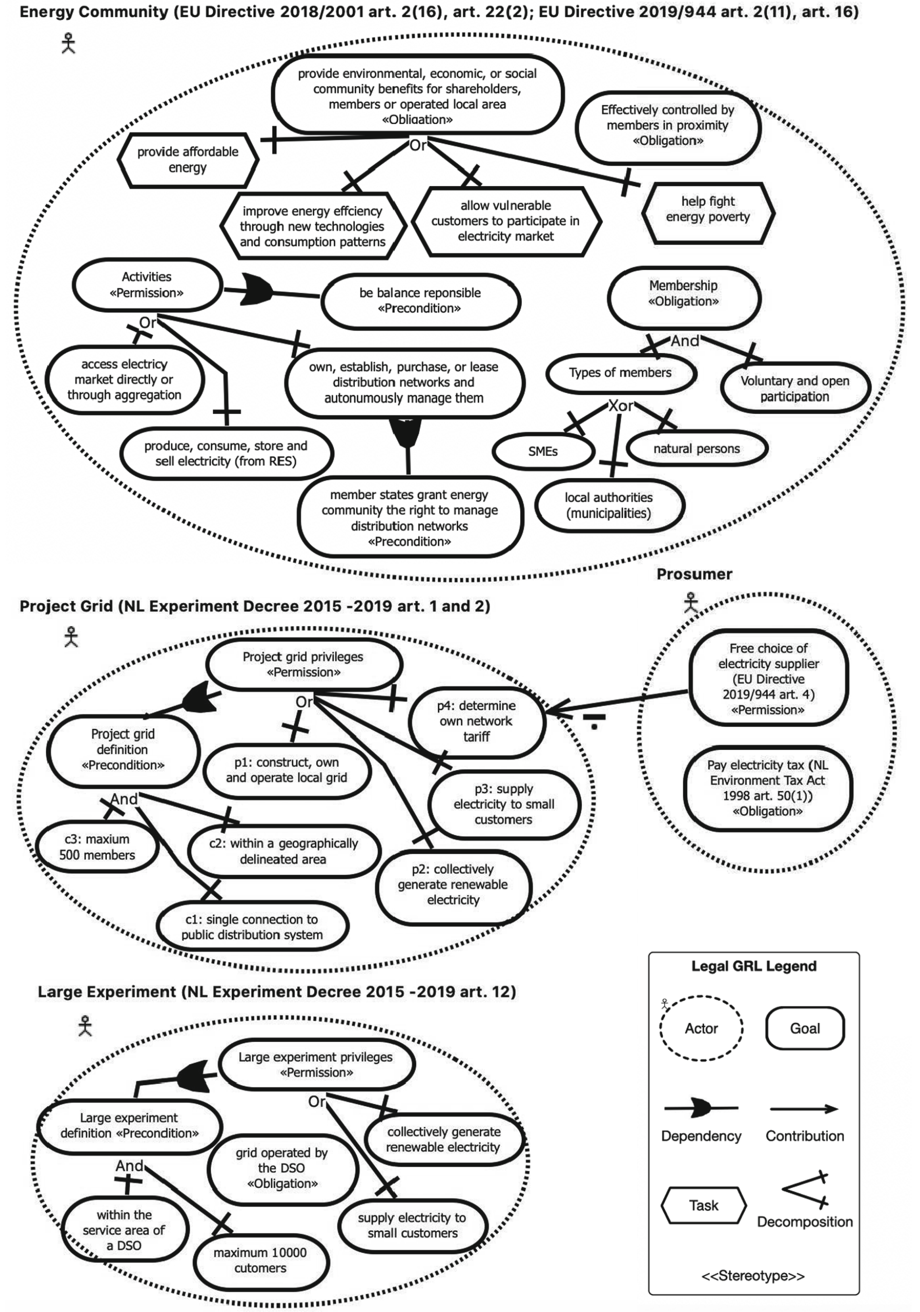


Fig. 2. Legal GRL model capturing extracts of regulations influencing the Schoonschip energy community [35]

Secondly, Article 50(1) of the Dutch National Environmental Tax Act states another provision regarding the consumption of electricity: "*...tax is levied on the supply of ... electricity via a (grid) connection to the consumer*". As the subject of this sentence is a passive object namely "tax", we rephrase it into active mode as "customers have to pay tax for all electricity supply via a grid connection". Then following the guidance offered by Legal GRL, we classify this (rephrased) statement as an «Obligation» goal in the Legal GRL model. This goal should be fulfilled by all the 46 prosumers connected to the local Schoonschip smart grid for exchanging self-generated electricity.

Thirdly, the Schoonschip community is a "citizen energy community" according to EU Directive 2019/944 Article 2(11). Therefore, according to EU Directive 2019/944, Article 16(2b): "*member states may provide ... that citizen energy communities are entitled to own, establish, purchase or lease distribution networks and to autonomously manage them subject to conditions set out in paragraph 4 of this Article;*", whereby paragraph 4 states "member states may decide to grant citizen energy communities the right to manage distribution networks in their area of operation ...". We model this statement as a «Permission» goal of Energy Community in the Legal GRL model. Moreover, this «Permission» goal can be satisfied only if the precondition outlined in the statement is fulfilled. We model the precondition of the statement as a «Precondition» goal and associate it with the «Permission» goal via a dependency link. Now, the Netherlands temporarily grants an energy community in the context of the Dutch regulatory sandbox — the Experiment Decree (hereafter Decree), the right to manage distribution networks, if the community is a "project grid". More specifically, Article 1 of the Decree sets forth the following conditions for project grids "(c1) have one connection to the public distribution system, (c2) are located within a geographically delineated area, and (c3) have a maximum of 500 connected customers", and Article 2 states that "project grids may (p1) autonomously construct, own, and operate the local grid", in addition to the other privileges, namely (p2) collectively generate renewable electricity, (p3) supply electricity to small customers, and (p4) determine own network tariff [35].

In the Legal GRL model, we create the «Permission» goal "Project grid privilege" to represent the privileges as allowed by the Decree. This goal has four child goals linked via OR-decomposition to capture the four privileges p1-p4 respectively. Similarly, we create the «Precondition» goal "Project grid definition" to represent the conditions that must be satisfied to be recognized as a project grid. This precondition goal has three child goals linked via AND-decomposition corresponding to the three conditions c1-c3 respectively. The dependency link from the «Permission» goal to the «Precondition» goal indicates that to enjoy the privileges of a project grid, the project grid definition must be fulfilled.

3.2 Step 2: Designing the Value Network

Different modeling languages exist for value network identification being e^3*value* [11], the Value Delivery Modeling Language (VDML, [1]), and Resource-Event-Agent (REA) [7]. While we aim here at illustrating the role that modeling can play and so, the particular criteria for language selection are of less importance, we adopt *e^3value* due to its mature methodological guidance, strong software tool support[3], and its earlier use in the electricity sector [10,18,31]. *e^3value* focuses on modeling who exchanges what of value with whom [11]. While the details of how to construct a value model are beyond the scope of this paper [11], one takes as a point of departure the core needs/ideas underlying the project at hand and proceeds to identify the actors involved. Then one identifies what actors exchange of value with each other (in terms of value objects), and how value exchanges are grouped (in terms of value interfaces grouping ports to which value objects are attached). Also, one distinguishes between individual actors, as profit-and-loss responsible organizational units, and market segments, as collections of actors that assign the same value to a value object.

Illustration: Figure 3 (a) presents the "ideal-type" Schoonschip value network in *e^3value*. Here, we observe that Schoonschip, as an organization, seeks to carry out three main value activities [35]: local energy sharing, offering demand-side flexibility, and operating a local grid (modeled with a dashed line). Local energy sharing concerns the coordination of green electricity (modeled as a value object) among prosumers, whereby the need for electricity of one prosumer is matched with the production of electricity of other prosumers. Offer demand side flexibility, meanwhile, concerns sharing the common capacity of prosumers as balancing services to so-called Balance Responsible Parties outside of Schoonschip. In particular, both storage capacity and green electricity are offered to actors that are responsible for preventing over- or under-supply of electricity on the grid. Finally, we observe the value activity operate a local grid, in which Schoonschip monitors and maintains its grid infrastructure, or at least holds the main responsibility for operating its own grid (possibly supported by parties of their own choice), supported by maintenance fees from the prosumers.

The above three value activities contribute to the ideal type aims of Schoonschip, namely, to be social/local through self-sufficiency in electricity production and consumption (through local energy sharing, as well as local grid operation), as well as to be sustainable (by means of offering green electricity to the local prosumers and balancing responsible parties). Nevertheless, as reported in [35], regulation influences both the extent to which these ideal type aims can be carried out and how these aims are to be carried out, as discussed next.

3.3 Step 3: Regulatory Analysis of the Value Network

Using the idealized value network as a point of departure, in this step we identify the regulation of relevance. We do so by first identifying focal actors and market segments in the value network, with focal actors being those that are core to realize the project at hand. Thereafter, we use the regulatory taxonomies discussed in Sect. 2 to focus on energy sector specific regulation and to, both, (1) tease out relevant information from the value model, as well as (2) guide the finding of supplemental information as needed for the selected regulation.

[3] https://www.thevalueengineers.nl/e3value-tools/.

Illustration: For the Schoonschip value network, both the prosumer and Schoonschip can be identified as focal actors. To guide the selection of the relevant regulations, we adapt the taxonomy dimensions from [3] and for our purposes arrive at size, geography, grid connection, key stakeholders, exemptions, and unit of analysis. Firstly, we consider these taxonomy dimensions in light of the value model from the previous step. As such, in terms of the dimension size, we know that 46 households act as prosumers, as expressed in terms of the size of the market segment. Secondly, we consider the taxonomy dimensions in terms of additional information gleaned from the documentation available on Schoonschip [35], such as Schoonschip being tied to a single area (on the dimension geography), and having a single connection to the grid (grid connection).

Based on the analysis, we classify the Schoonschip as a "Project grid" (Fig. 2), with the dimensions size, geography, and grid connection satisfying the three leaf goals for its precondition "Project grid definition'. Interestingly, if a project scores differently on the *same* dimensions of size, geography, and grid connection, in the Netherlands the project could also be classified as an experimentation grid. In particular, returning to the modeled regulation in Fig. 2 observe that for a project to fall under a "Large experiment definition" it must satisfy the leaf goals of a "maximum 10000 customers" (size), and to be within the service area of a DSO (geography). So here one can observe how such taxonomies can be helpful, not only in navigating models but also in comparison of different regulations to tease out which one is applicable. Additionally, for prosumers as focal actors we know that regulations for typical households are applicable, being Fig. 2 the "Free choice of electricity supplier (EU Directive 2019/944 Art. 4)", as well as "Pay electricity tax (Environmental Electricity Act Art. 50 (1))".

3.4 Iteration of Step 2: (Re-)designing the Value Network

Now we return to the step of designing the value network and consider explicitly the effect which the selected regulation has. In so doing, we consider the different implications regulation can have for value network design, being (cf. Challenge 1a-1c): (1) the changing role of an actor, (2) a change in the distribution of costs and benefits, (3) the realization of ideal type benefits.

Illustration: We find that the value network modeled initially in Step 2 should be revised in three ways, as annotated by the three labels in Fig. 3 (b). Firstly, as per label ①, the regulation "Free choice of electricity supplier (EU Directive 2019/944 Art. 4)" of the prosumer affects the value network by introducing a choice on the side of the prosumer between, on the one hand, consuming green electricity from Schoonschip, and, on the other hand, consuming electricity from an external electricity supplier. Since the economic viability of Schoonschip and its prosumers to a large extent depends on exchanging electricity within Schoonschip, it is important to account for this free choice in the value model, especially by varying the amount of external electricity consumed, and analysing its effect on the overall economic feasibility of Schoonschip.

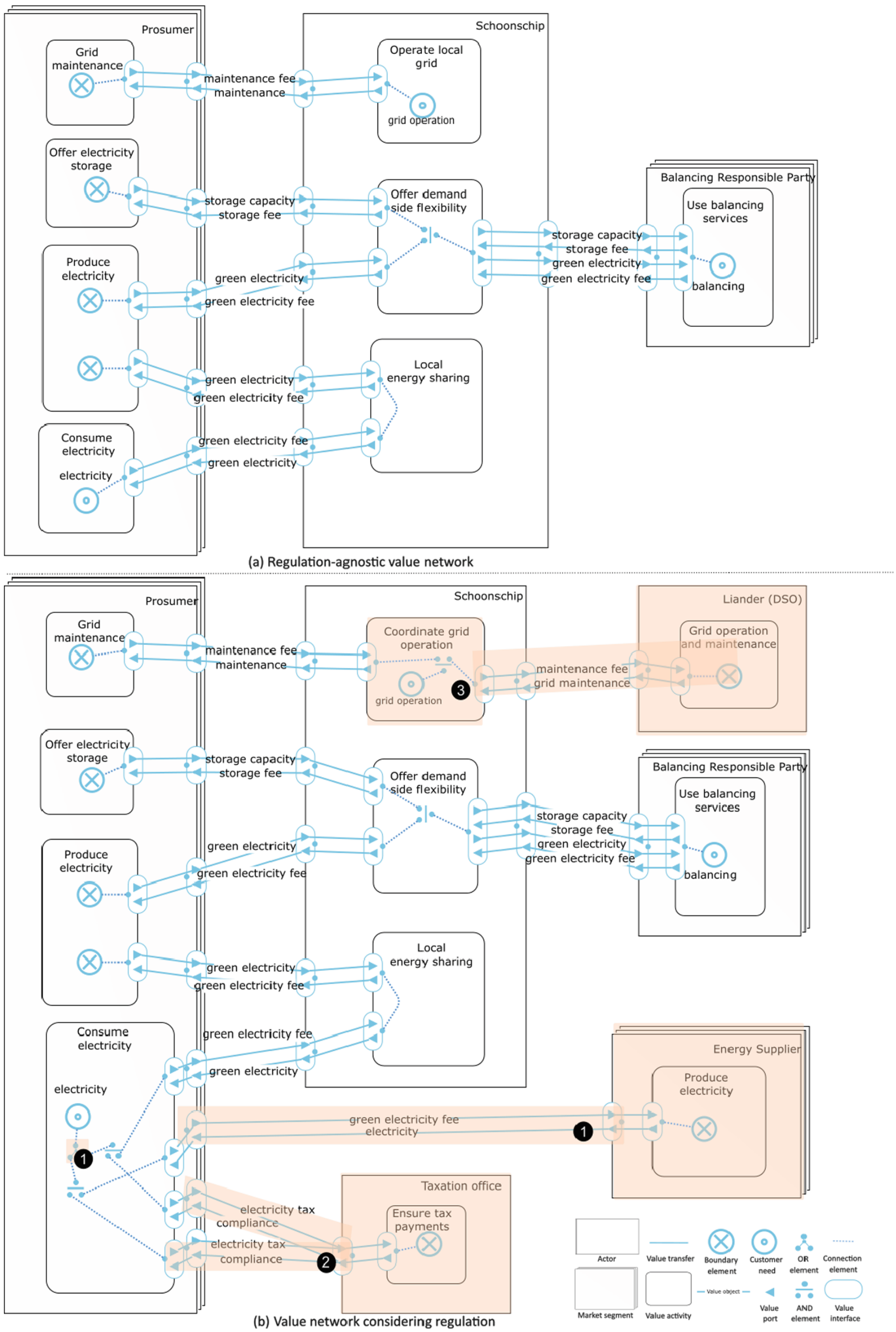


Fig. 3. The regulation-agnostic value network for Schoonschip (in (a)), and its regulatory counterpart (in (b)), considering regulation on (1) free choice of energy supplier, (2) taxation, (3) the role of the DSO

Secondly, as per label ②, the regulation "Pay electricity tax (Environmental Electricity Act Art. 50 (1))" implies that the prosumer has to pay taxes anytime electricity is exchanged (both within Schoonschip and when electricity is sourced externally). Such tax payments also affect the economic feasibility of Schoonschip, hence should be modeled. Nevertheless, the fact that electricity taxes have to be paid in energy communities is also an interesting (negative) economic impact of the current regulations in the Netherlands, which provide no special exemptions for energy communities. Differently, when a similar community would be set up in a different member state, for example in the Brussels region of Belgium[4], incentives are catered for on taxes and surcharges to be paid on electricity exchanged within energy communities.

Finally, and thirdly, as per label ③, the regulation "construct, own and operate own grid" from the project grid decree also has interesting implications for the value network. Namely, initially, the idea of Schoonschip, as per Step 2, was to operate its own local grid, which is very much in line with the project grid decree. Nevertheless, in 2019 the Dutch government decided to abandon the regulatory sandbox regulations that the project grid decree is part of [35]. The abandonment of the project grid decree also has implications for the design of the value network. Namely, now falling under conventional Dutch regulation the local grid infrastructure needs to be operated and maintained by Liander, the local DSO which takes care of grid infrastructure in the region where Schoonschip is located. Thus Schoonschip has to pay an additional actor for grid operation, the DSO, which it would ideally be responsible for itself.

4 Discussion and Next Steps

As we have shown in the previous section, the application of conceptual modeling may support the development of energy communities when it comes to being able to (1) systematically cater for existing regulations and resulting obligations for different actors (see the Legal GRL models), and (2) explicitly representing the value network and related exchanges, and in that way partly catering for Challenges 1 and 2. By relying on conceptual models, we capitalize on their capability to foster communication and understanding, thus making the design of energy communities more accessible to end users, cf. Challenge 3, see also [18]. The communication capabilities are fostered not only by the main feature of models to help handle the complexity through abstraction but also by their visual nature and dedicated concepts offered, see, e.g., the goal models showing the regulatory setting, Fig. 2. Those features of conceptual models shall not only foster communication during the energy community design process but also (ideally) leverage a shared understanding among stakeholders.

However to account for peculiarities of energy communities, the changing role of actors, or showing exactly the influence of the existing regulations, cf. Sect. 2, the support provided by existing modeling languages is limited. This is partly the case as (1) supporting the targeted analyses is not an explicit aim

[4] https://www.rescoop.eu/policy/belgium-brussels.

of the selected modeling languages, and (2) they have not been proposed in the context of the electricity sector, and therefore, provide concepts applicable to any domain. When it comes to the latter, it has its advantages and disadvantages. On the one hand, the application of generic terminology makes the model more understandable to a wide audience (not necessarily being electricity sector experts), on the other hand, this can cause issues when trying to account for domain-specific phenomena, such as actors specific to the electricity sector and their obligations. In addition, although the application of Legal GRL and e^3*value* clearly supports the targeted assessment, and as such, offers utility to interested stakeholders, please note that the created models constitute separate artifacts, as there is a lack of integration between the two languages and tools supporting them. As a result, relations between those models (e.g., the specific impact of the regulation on the specific aspect of a value network) need to be established by the person(s) conducting the analysis. Thus, substantial part of the analysis is carried out 'outside' of the created models and depends on the knowledge and competencies of the human actor involved.

Considering the above, we argue that there is a clear need for a dedicated modeling method encompassing a dedicated domain-specific modeling language (or languages) (DSML) offering an integrated perspective on all relevant aspects of energy communities, and a dedicated process model with a set of heuristics and guidelines accounting for the existing knowledge and ensuring compliance with the regulatory aspects. Firstly, such a *dedicated domain-specific modeling language* should account for aspects specific to energy communities and dependencies between these aspects, among others as captured in already mentioned taxonomies, starting with relevant regulations and policies, through internal functioning and success conditions, to micro-grid specification and location. Embedding the relevant domain-specific knowledge into the language specification comes with different advantages, among them (1) supporting the integrity and fostering the quality of the modeling result, but also (2) fostering modeling productivity, in the sense that the relevant knowledge is embedded into the language and does not have to be each time developed from scratch, see [6]. Please also note that this taxonomy can guide, both, the development of regulatory models specific to energy communities, as well the use of the modeling method for selecting relevant regulations. We may use the aspects mentioned in the taxonomy to filter out relevant information from the corresponding model (e.g., for a value model, among others, the size of the prosumer market segment, also responsible actors), and supplement it with additional information needed (geographic proximity, single connection). Secondly, such a DSML should provide an integrated perspective and allow considering the regulatory and business development in tandem, e.g., by clearly showing what value exchanges are influenced (and how) by existing regulations. Finally, such a DSML should address the deficits of currently available languages, e.g., by differentiating between actors and roles, or by being able to account for different types of legal entities, see Sect. 3.

Although existing methods already provide useful guidance when designing an energy community, we argue that *a dedicated process model* tailored to the specifics of energy communities wherein the modeling of business case development and regulation are closely intertwined is needed. While the exact shape of such a process model is open for further work, we consider taking inspiration from e3control, see [17], where for the needs of compliance checking and distribution of relevant knowledge, patterns are used. We envision patterns to account for (1) context (e.g., regulations applicable in the given region at some particular point in time), problem (e.g., the definition of value exchanges among involved actors), and solution (e.g., typical/allowed configuration of actors and relevant value exchanges). Such patterns may not only be used to support compliance checking and compliance analysis [2], but they also support decision-making, e.g., to decide on the regulatory suitability of several value networks for deploying an energy community, or on the distribution of costs and benefits over actors according to the obligations and rights granted by a regulation.

5 Conclusions

In this paper, we focus on the support of conceptual modeling for a conjoint regulatory and viability analysis. Based on a realistic scenario and the conjoint usage of the two selected modeling languages, we show the added value resulting from the application of models but also indicate the need for a dedicated modeling method.

When it comes to limitations, firstly, while our case is based on a real, well-documented scenario, we in the end rely on secondary sources. As a result, we have somewhat simplified value models and goal models. In addition, we have used two selected languages only, whereas more languages could have been used, for example, to cover further project development aspects. Finally, the modeling of the selected case allowed us to show mainly the extent to which Challenges 1 & 2 can be addressed. Showing that the application of conceptual modeling indeed may foster understanding and communication in this particular case would require involvement and/or interactions with relevant stakeholders.

References

1. Value delivery modeling language 1.1 (2018). https://www.omg.org/spec/VDML/1.1/PDF
2. Akhigbe, O., Amyot, D., Richards, G.: A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance. RE **24**(4), 459–481 (2019)
3. Bovera, F., Lo Schiavo, L.: From energy communities to sector coupling: a taxonomy for regulatory experimentation in the age of the European green deal. Energy Policy **171**, 113299 (2022)
4. Cambini, C., Congiu, R., Jamasb, T., Llorca, M., Soroush, G.: Energy systems integration: implications for public policy. Energy Policy **143**, 111609 (2020)
5. Diestelmeier, L., Swens, J.: Energy communities in the Netherlands: learning from local energy initiatives. In: European Energy Law Reports XIV, pp. 239–264 (2021)

6. Frank, U.: Domain-specific modeling languages: requirements analysis and design guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) Domain Engineering. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-36654-3_6
7. Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-rea enterprise information architecture. IJAIS **3**(1), 1–16 (2002)
8. Ghanavati, S.: Legal-URN framework for legal compliance of business processes. Ph.D. thesis, University of Ottawa (2013)
9. Giordano, V., Onyeji, I., Fulli, G., Jimenez, M.S., Filiou, C.: Guidelines for conducting a cost-benefit analysis of smart grid projects. Technical Report EUR 25246 EN (2012)
10. Gordijn, J., Akkermans, H.: Business models for distributed generation in a liberalized market environment. Electr. Power Syst. Res. **77**(9), 1178–1188 (2007)
11. Gordijn, J., Akkermans, J.: Value-based requirements engineering: exploring innovative e-commerce ideas. Requirements Eng. **8**, 114–134 (2003)
12. Hohfeld, W.N.: Fundamental legal conceptions as applied in judicial reasoning. Yale Law J. **26**(8), 710–770 (1917)
13. Hotie, F., Gordijn, J.: Value-based process model design. Bus. Inf. Syst. Eng. **61**(2), 163–180 (2019)
14. Hubmann, G.: The socio-spatial effects of circular Urban systems. In: IOP Conference Series: Earth and Environmental Science, vol. 1078. IOP Publishing (2022)
15. Inês, C., Guilherme, P.L., Esther, M.G., Swantje, G., Stephen, H., Lars, H.: Regulatory challenges and opportunities for collective renewable energy prosumers in the EU. Energy Policy **138**, 111212 (2020)
16. Joint Research Centre: Harmonised system-wide cost-benefit analysis for candidate smart electricity grid projects. Technical report, European Commission (2022)
17. Kartseva, V., Gordijn, J., Tan, Y.: Value-based design of networked enterprises using e^3control patterns. In: RE 2007, pp. 91–100. IEEE (2007)
18. de Kinderen, S., Kaczmarek-Heß, M., Ma, Q., Razo-Zapata, I.S.: Model-based valuation of smart grid initiatives: foundations, open issues, requirements, and a research outlook. Data Knowl. Eng. **141**, 102052 (2022)
19. de Kinderen, S., Ma, Q., Kaczmarek-Heß, M., Eshuis, R.: Beyond business development: regulatory assessment of energy sector projects with contextual requirements engineering. In: Barn, B.S., Sandkuhl, K. (eds.) PoEM 2022. Lecture Notes in Business Information Processing, vol. 456. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-21488-2_12
20. Krug, M., et al.: Comparative assessment of enabling frameworks for RECs and support scheme designs (2022). https://doi.org/10.5281/zenodo.7622527
21. Lode, M.L., Heuninckx, S., Te Boveldt, G., Macharis, C., Coosemans, T.: Designing successful energy communities: a comparison of seven pilots in Europe applying the multi-actor multi-criteria analysis. ERSS **90**, 102671 (2022)
22. Lowitzsch, J., Hoicka, C.E., van Tulder, F.J.: Renewable energy communities under the 2019 European clean energy package-governance model for the energy clusters of the future? Renew. Sustain. Energy Rev. **122**, 109489 (2020)
23. Luthra, S., Kumar, S., Kharb, R., Ansari, M.F., Shimmi, S.: Adoption of smart grid technologies: an analysis of interactions among barriers. Renew. Sustain. Energy Rev. **33**, 554–565 (2014)
24. Meitern, M.: Creating an energy community: what do you need to know? (2022). https://www.renaissance-h2020.eu/creating-an-energy-community-what-do-you-need-to-know/

25. Mihailova, D., Schubert, I., Burger, P., Fritz, M.M.: Exploring modes of sustainable value co-creation in renewable energy communities. J. Clean. Prod. **330**, 129917 (2022)
26. Moroni, S., Alberti, V., Antoniucci, V., Bisello, A.: Energy communities in the transition to a low-carbon future: a taxonomical approach and some policy dilemmas. J. Environ. Manage. **236**, 45–53 (2019)
27. Muthuri, R., Capecchi, S., Sulis, E., Amantea, I.A., Boella, G.: Integrating value modeling and legal risk management: an IT case study. Inf. Syst. e-Bus. Manag. **20**, 1–29 (2022)
28. Mylopoulos, J.: Conceptual modelling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modelling, Databases, and CASE: an Integrated View of Information System Development. Wiley, Hoboken (1992)
29. Niesten, E., Alkemade, F.: How is value created and captured in smart grids? A review of the literature and an analysis of pilot projects. Renew. Sustain. Energy Rev. **53**, 629–638 (2016)
30. Nordling, A., Pädam, S., af Burén, C., Jörgensen, P.: Social costs and benefits of smart grid technologies (2018)
31. Poels, G., Kaya, F., Verdonck, M., Gordijn, J.: Early identification of potential distributed ledger technology business cases using e^3value models. In: Advances in Conceptual Modeling (ER 2019 Workshop). LNCS, vol. 11787, pp. 70–80 (2019)
32. Reis, I.F., Gonçalves, I., Lopes, M.A., Antunes, C.H.: Business models for energy communities: a review of key issues and trends. Renew. Sust. Energ. Rev. **144**, 111013 (2021)
33. Schittekatte, T., Meeus, L., Jamasb, T., Llorca, M.: Regulatory experimentation in energy: three pioneer countries and lessons for the green transition. Energy Policy **156**, 112382 (2021)
34. Strecker, S., Heise, D., Frank, U.: RiskM: a multi-perspective modeling method for it risk assessment. Inf. Syst. Front. **13**, 595–611 (2011)
35. Swens, J., Diestelmeier, L.: Developing a legal framework for energy communities beyond energy law. In: Energy Communities, pp. 59–71. Academic Press (2022)
36. Wieringa, R.J.: Design Science Methodology for Information Systems and Software Engineering. Springer, Cham (2014)

Author Index

H. A. Proper et al. (Eds.): EDOC 2023, LNCS 14367, pp. 223–224, 2024.
https://doi.org/10.1007/978-3-031-46587-1

T

V

W